Decision Support for Response Selection in Maritime Law Enforcement

Alexandre Colmant

Dissertation presented for the degree of
Doctor of Philosophy
in the Faculty of Engineering at Stellenbosch University

Promoter: Prof JH van Vuuren

December 2016

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December, 2016

Abstract

In the context of *maritime law enforcement* (MLE), a human operator is typically required to make a variety of counter-threat decisions following the detection and evaluation of threats at sea. These decisions reside within a so-called *response selection* process during which MLE resources, such as patrol vessels, military vessels and armed helicopters, have to be dispatched to intercept maritime vessels that are deemed potential threats. Because the number of potential maritime threats can be overwhelming and the nature of the decision process is typically complex, the quality of resource assignment decisions can be improved significantly by providing maritime operators with computerised decision support.

A generic, semi-automated MLE response selection and resource routing *decision support system* (DSS) is designed in this dissertation. This DSS is capable of assisting human operators in spatio-temporal resource dispatch decision making so that MLE resources may be employed effectively and efficiently. These decisions are made based on kinematic vessel-related data obtained from associated threat detection and threat evaluation systems, as well as subjective input data contributed by MLE response selection operators. Fully automated decision making is therefore not pursued; the aim of this study is to establish a support tool for an operator.

Multiple response selection objectives are accommodated in the proposed DSS in a generic manner, so as to provide users of the system with the freedom of configuring their own, preferred goals. This generic DSS design is populated with examples of models capable of performing the functions of the various constituent parts of the system, and their workability is tested and demonstrated by solving the MLE response selection problem in the context of two realistic, but simulated, MLE scenarios.

The MLE DSS proposed in this dissertation may be used in future to assist maritime operators in their complex decision making processes. In particular, operators may use it as a guideline to validate and/or justify their decisions, especially when the level of uncertainty pertaining to the observed maritime scenario is high and/or only parts of the problem may be resolved by hand. Use of this system in a real-world context is expected to reduce the stress levels of operators typically associated with difficult decisions, while simultaneously improving the overall quality of MLE decisions in an integrated fashion.

Uittreksel

In die konteks van *maritieme wetstoepassing* (MWT) word daar tipies van 'n menslike operateur verwag om, in reaksie op die opsporing en evaluering van maritieme bedreigings, 'n verskeidenheid besluite te neem waarvolgens hierdie bedreigings geneutraliseer kan word. Hierdie besluite vind plaas binne 'n sogenaamde *MWT-responsseleksieproses* waartydens MWT-hulpbronne, soos patrolliebote, militêre skepe en gewapende helikopters, ontplooi moet word om maritieme vaartuie wat as bedreigings beskou word, te onderskep. Aangesien die aantal potensiële maritieme bedreigings oorweldigend kan wees en die aard van die MWT-besluitnemingsproses tipies kompleks is, kan die kwaliteit van hulpbrontoedelingsbesluite noemenswaardig verhoog word deur maritieme operateurs met gerekenariseerde besluitsteun te bedien.

'n Generiese, gedeeltelik ge-outomatiseerde *besluitsteunstelsel* (BSS) word in hierdie dissertatie vir MWT-responsseleksie en hulpbronroetering ontwerp. Hierdie BSS is daartoe in staat om menslike operateurs met effektiewe en doeltreffende besluitsteun in terme van die ontplooiing van MWT-hulpbronne oor tyd en ruimte te bedien. Hierdie besluite word gemaak gebaseer op kinematiese vaartuigdata wat vanuit gepaardgaande BSSe vir bedreigingsopsporing en bedreigingsafskatting verkry word, sowel as subjektiewe toevoerdata vanaf MWT-responsseleksie operateurs. Ten volle ge-outomatiseerde besluitneming word dus nie nagestreef nie; die doel van hierdie studie is om 'n steunstelsel vir 'n menslike operateur daar te stel.

'n Verskeidenheid responsseleksiedoele word in die voorgestelde BSS op 'n generiese wyse geak kommodeer om sodoende aan gebruikers die vryheid te bied om hul eie voorkeurdoele te spesifiseer. Hierdie generiese BSS-ontwerp word vergesel met voorbeelde van modelle waarvolgens die verskeie funksionaliteite van die onderskeie stelselkomponente gerealiseer kan word, en hulle werkbaarheid word aan die hand van twee realistiese, maar gesimuleerde, MWT-scenarios getoets en gedemonstreer.

Die MWT BSS wat in hierdie dissertatie voorgestel word, mag in die toekoms gebruik word om maritieme operateurs in hulle komplekse besluitprosesse by te staan. In die besonder kan operateurs die stelsel as 'n riglyn gebruik om hul besluite te valideer of te regverdig, veral wanneer die vlak van onsekerheid onderliggend aan die maritieme scenario hoog is en/of slegs dele van die MWT-responsseleksieprobleem met die hand oplosbaar is. Daar word verwag dat gebruik van hierdie slelsel in die praktyk die stresvlakke van operateurs as gevolg van moeilike MWT-besluite noemenswaardig kan verlaag, en terselfdertyd die oorkoepelende kwaliteit van MWT-besluite op 'n geïntegreerde wyse kan verbeter.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- My promoter, Prof Jan van Vuuren, for his dedication, guidance and valuable insight throughout the compilation of this dissertation. I admire his professionalism and inclination towards promoting a standard of excellence in the work delivered. I appreciate his enthusiasm towards the nature of the work pursued in this dissertation, as well as the numerous intellectual conversations shared with him during the past three years.
- My colleague, Jacques du Toit, for his valuable insight into the maritime law enforcement domain, particularly in the early stages of work towards this dissertation.
- The Departments of Logistics and the SUnORE group for the use of their excellent computing facilities, office space and kitchens.
- The Armaments Corporation of South Africa (ARMSCOR) for funding the research reported in this dissertation.
- Finally, my friends and family for their moral support and encouragements during the past three years, and their understanding during times of great pressure and unavailability to socialise on my part.

Table of Contents

List of Reserved Symbols

The symbols listed below are reserved for a specific use. Other symbols may be used throughout the dissertation in an unreserved fashion.

Symbols in this dissertation conform to the following font conventions:		
$\mathcal{A}$	Symbol denoting a **set**	(Calligraphic capitals)
$\boldsymbol{a}, \boldsymbol{A}$	Symbol denoting a **vector** or **matrix**	(Boldface lower case letters or capitals)
a, A	Symbol denoting a **parameter** or **variable**	(Roman lower case letters or capitals)

Symbol	Meaning
Fixed parameters	
$A^d_{k\rho}$	The distance autonomy threshold of MLE resource k for patrol circuit ρ
$A^t_{k\rho}$	The time autonomy threshold of MLE resource k for patrol circuit ρ
C	The number of epochs in the method of simulated annealing
C^d_k	The trajectory disruption cost associated with MLE resource k
C^s_k	The setup cost incurred when preparing MLE resource k for departure on a mission
G_s	The initial generation size of subspecies s in the NSGA-II
O_{max}	The maximum number of attempts allowed in the crossover procedure of the NSGA-II
p_{brd}	The probability of performing a between-route delete-and-insert transformation in the method of simulated annealing
p_{brs}	The probability of performing a between-route swap transformation in the method of simulated annealing
p_{is}	The probability of performing an inter-state transformation in the method of simulated annealing
p_m	The mutation rate in the NSGA-II
p_{wr}	The probability of performing a within-route swap transformation in the method of simulated annealing
P_{size}	The population size in the NSGA-II
Q_h	The score associated with the neutralisation of class h threats
S^t_{kh}	The expected service time of MLE resource k for VOIs of threat class h
T_c	The temperature during epoch c in the method of simulated annealing
T^w_k	The expected setup time incurred when preparing MLE resource k for a mission
W_{kh}	The score associated with the efficiency of MLE resource k in terms of neutralising a class h threat

Z_s	The weight representing the relative importance of decision entity s to the coastal nation
α	A user-defined parameter for inter-state transformations in the method of simulated annealing
β	A user-defined parameter for inter-state transformations in the method of simulated annealing
β_{bk}	A binary parameter taking the value 1 if MLE resource k is allowed to be scheduled to end its route at base b, or zero otherwise
δ_{kh}	A binary parameter taking the value 1 if MLE resource k is strictly incapable of neutralising VOIs of threat class h, or zero otherwise
η_k	The fixed average speed of MLE resource k
Γ_k	The cost per kilometre travelled associated with MLE resource k
λ	A user-defined parameter for generating the initial population in the NSGA-II
μ	A user-defined parameter for generating the initial population in the NSGA-II
π	The cooling parameter in the method of simulated annealing
σ	A user-defined parameter for generating the initial population in the NSGA-II
θ_{kh}	An upper bound threshold associated with MLE resource k and threat type h for managing the risk of infeasible encounters
Dynamic parameters	
$a^d_{k\tau}$	The distance autonomy level of MLE resource k at the start of time stage τ
$\tilde{a}^d_{k\tau}$	The estimated distance autonomy level of MLE resource k as soon as it has finished servicing the last VOI along its visitation route during time stage τ
$a^t_{k\tau}$	The time autonomy level of MLE resource k at the start of time stage τ
$\tilde{a}^t_{k\tau}$	The estimated time autonomy level of MLE resource k as soon as it has finished servicing the last VOI along its visitation route during time stage τ
$C^e_{ikh\tau}$	The estimated cost resulting from a potentially infeasible encounter between VOI i and MLE resource k during time stage τ
$d_{ijk\tau}$	The length of travel arc $(i,j)_\tau$ during time stage τ
n_τ	The number of VOIs tracked during time stage τ
$N_{s\tau}$	The ideal number of VOIs that decision entity s would prefer to investigate during time stage τ
$p_{ih\tau}$	The probability that VOI i resides in threat class h during time stage τ
$S_{ik\tau}$	The expected service time of MLE resource k for VOI i during time stage τ
$S'_{i\tau}$	The expected remaining time required to complete the service of VOI i at the end of time stage τ
$T^d_{i\tau}$	The time elapsed between the detection of VOI i and the start of time stage τ
Δ_τ	The minimum proportion of VOIs that have to be intercepted during time stage τ
$\gamma_{k\tau}$	A binary parameter taking the value 1 if MLE resource k is idle and not in patrol mode at the end of time stage $\tau-1$, or zero otherwise
Sets	
$\mathcal{A}$	The archived set of non-dominated solutions in the method of simulated annealing
$\mathcal{E}_{k\tau}$	The VOI exclusion set associated with MLE resource k during time stage τ
$\mathcal{F}$	The set of objective function vectors in the Pareto front
$\tilde{\mathcal{F}}$	The set of objective function vectors in the non-dominated front
$\mathcal{H}$	The set of indices of VOI threat types

$\mathcal{I}_{k\tau}$	The VOI unordered inclusion set associated with MLE resource k during time stage τ
$\widehat{\mathcal{I}}_{k\tau}$	The VOI ordered inclusion set associated with MLE resource k during time stage τ
$\mathcal{O}_{s\tau}$	The ordered set of input data received from decision entity s at the beginning of time stage τ
$\mathcal{P}$	The set of Pareto optimal solutions
$\mathcal{P}_{\boldsymbol{x}}$	The portion of $\mathcal{P}$ that dominates the solution $\boldsymbol{x}$
$\tilde{\mathcal{P}}$	The set of candidate solutions whose corresponding objective function vectors are elements of $\tilde{\mathcal{F}}$
$\mathcal{V}^e_\tau$	The set of indices of VOIs at the start of time stage τ
$\mathcal{V}^r$	The set of indices of MLE resources
$\mathcal{V}^r_{0_\tau}$	The set of indices of MLE resource initial spatial locations
$\mathcal{V}^b$	The set of indices of bases
$\mathcal{V}^b_{k\tau}$	The set of bases to which MLE resource k may travel after completing its mission during time stage τ
$\mathcal{V}^r_{c\tau}$	A subset of MLE resources configured in a convoyed assignment during time stage τ
$\mathcal{V}^p$	The set of indices of patrol circuits
$\mathcal{V}^p_{k\tau}$	The set of patrol circuits to which MLE resource k may be assigned after completing its mission during time stage τ
$\mathcal{V}^e_{k\tau}$	The ordered set of VOIs scheduled to be investigated by MLE resource k during time stage τ
$\mathcal{Z}$	The set of indices of decision entities

Variables	
$x_{ijk\tau}$	A binary variable taking the value 1 if MLE resource k is scheduled to traverse arc $(i,j)_\tau$ during time stage τ, or zero otherwise
$y_{ik\tau}$	A binary variable taking the value 1 if MLE resource k is scheduled to visit VOI i during time stage τ, or zero otherwise
$z_{is\tau}$	A binary variable taking the value 1 if VOI i is scheduled on a route assigned to entity s during time stage τ, or zero otherwise
$w^d_{k\rho\tau}$	A linking constraint variable
$w^t_{k\rho\tau}$	A linking constraint variable

Functions	
$E(\boldsymbol{x})$	The energy of solution $\boldsymbol{x}$ in the method of simulated annealing
f_c	A discrete function measuring the penalty incurred from the spread around $N_{s\tau}$ in an intermediate decision making paradigm
f_c^+	A discrete function measuring the penalty incurred from the spread around $N_{s\tau}$ in the decentralised decision making paradigm
f^o_{norm}	A normalising function along the values of objective o
$f(\boldsymbol{x})\|o$	The value of objective function o achieved by a solution $\boldsymbol{x}$
f_M	The marker function in the crossover operator of the NSGA-II
H^i	The hypervolume indicator
$s(H^i)$	The hypervolume indicator standard deviation

List of Acronyms

AMOSA: Archived Multiobjective Simulated Annealing

DSS: Decision Support System

FNSA: Fast Non-dominated Sorting Algorithm

HMI: Human Machine Interface

HSO: Hypervolume by Slicing Objectives

MLE: Maritime Law Enforcement

NSGA-II: Non-dominated Sorting Genetic Algorithm II

UNCLOS: United Nations Convention on the Law of the Sea

VOI: Vessel of Interest

VRP: Vehicle Routing Problem

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

Following the raid of an al-Qa'ida intelligence base in May 2011, evidence was uncovered suggesting that the terrorist organisation planned to blow up an unknown number of oil tankers within the US maritime transportation system [146]. Due to the strengths of their hulls, these tankers either have to be blown up from the inside (as the result of an inside-job) or, more likely, have to be intercepted by a terrorist vessel while in transit (and then blown up). The large quantity of tankers transiting these waters, combined with the vastness of the US maritime transportation system and the limited security resources available, however, has made it impossible for the US maritime defense authorities to achieve constant coverage of potentially threatening activities occurring in the vicinity of these terrorist targets. In addition, it was believed that the adversary had had the opportunity to observe security patrol patterns in order to better plan its attacking strategy. This alarming situation, coupled with challenging economic times, has forced the US maritime defense authorities to deploy its resources as effectively as possible so as to minimise the risk of allowing potentially threatening activities to progress towards the destruction of one or more targets, while simultaneously attempting to maintain operating costs at a minimum level [148].

1.1 Background

The obligations and privileges of coastal nations form part of a global legislation process to create an efficient international framework defining their rights and responsibilities in respect of seabeds and ocean floors. The establishment of such a framework is, however, a very complex process, and has for centuries been a frustrating and unfulfilled idealistic notion [151]. It is instructive to review certain laws pertaining to activities at sea in order to establish a context for the complex challenges faced by coastal nations.

1.1.1 The law of the seas

In 1982, the *United Nations Convention on the Law of the Sea* (UNCLOS) at last settled on a legal order for the seas and oceans with the aim of promoting international communication, peaceful uses of the seas, equitable and efficient utilisation of maritime resources, jurisdiction demarcation over waters, and conservation of maritime ecosystems [49, 124, 151]. In addition, this order clearly defines the nature of crimes committed at sea and provides coastal nations with the appropriate responses to enforce a variety of laws aimed at curbing these crimes. Under the UNCLOS, every coastal nation is given the rights and responsibilities to establish territorial waters over which they exercise various controls, as follows:

- The outer limit of the *territorial sea* of a coastal nation is the line consisting of every point at a distance of twelve nautical miles from the nearest point along the coastline [151]. The standard baseline for measuring the width of the territorial sea is the low-water line along the coast, as marked on large-scale maps officially recognized by the coastal nation[1]. Within this territory, the coastal state may exercise all sovereign rights over the seabed, water and associated airspace, but is obliged to afford the right of innocent passage to vessels of any state.
- The waters contiguous to the territorial sea, known as the *contiguous zone*, extends for twenty four nautical miles from the same baseline from which the extent of the territorial sea is measured. Here, the coastal state may exercise the rights to prevent violations of its fiscal, customs, immigration and waste disposal laws [151]. Additionally, the coastal state may also take actions within this zone to punish violations of these laws previously committed within its land territory or territorial sea.
- The *exclusive economic zone (EEZ)* stretches out to sea for a further 188 miles from the seaward boundary of the territorial sea. Here, the coastal state has the sovereign rights to explore, exploit, conserve and manage living and nonliving natural resources; to establish and use artificial islands, installations, and structures; to conduct marine scientific research; and to protect and preserve the marine environment [151]. The UNCLOS is, however, in the process of allowing certain coastal nations to extend specific zones within their EEZ beyond 200 nautical miles from the low-water line along the coast, based on certain physical characteristics of the continental shelf [124].

Lastly, the waters beyond the EEZ (which are not considered archipelagic waters) are defined as the *high seas*. The sense of liberty enjoyed by seafarers in these waters is applicable to all states, whether coastal or not. Subject to certain rules and regulations laid down by the UNCLOS and other entities, navigators of the high seas have the right to freedom of navigation; the freedom to lay underwater infrastructure; the freedom to construct artificial islands, installations and structures; the freedom of fishing; and the freedom to perform scientific research [151]. It is, nevertheless, imperative that these waters remain crime-free and that activities in these regions are only aimed toward peaceful ends. The four general maritime zones described above are illustrated in Figure 1.1.

1.1.2 Activities of vessels at sea

Maritime activities by seafaring vessels embody a very important part of global society; the globally connected economy relies on the seas and adjoined littorals for fishing, access to natural

[1] In the case of islands situated on atolls or of islands having fringing reefs, the baseline for measuring the extent of the territorial sea is the seaward low-water line of the reef.

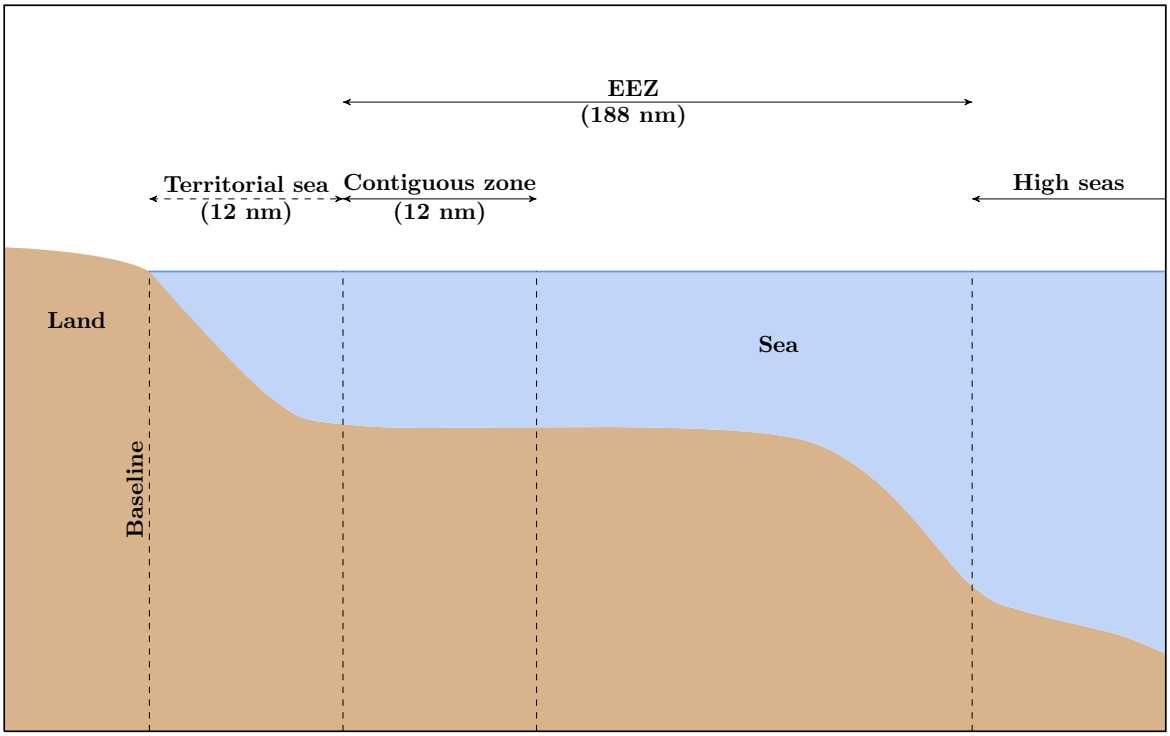


FIGURE 1.1: *The four maritime zones defined by the UNCLOS [151].*

resources and the transportation of most of the world's import and export commodities. These maritime activities today contribute, *inter alia*, to over 90 percent of global trade [8] and are directly responsible for 91 million tons of food for human consumption annually [39]. Effective governance of these maritime regions is therefore essential for both the economic growth and the national security of coastal nations. Indeed, according to the Maritime Security Sector Reform [124], "the maritime sector is fundamental, directly or indirectly, to the national defense, law enforcement, social, and economic goals and objectives of nearly every country. It is a crucial source of livelihood for many in developing nations, a platform for trade, and a theater for potential conflict or crime."

A sizable portion of maritime activities are unfortunately responsible for a wide variety of problems, ranging from being detrimental to only a few individuals to harming society on a global scale. These maritime problems are typically caused by lawless vessels that choose to disrupt the harmony at sea for personal gain. Such activities, or *threats*, typically include piracy acts, illegal or unregulated fishing, narcotics trafficking, illegal immigration, environmental degradation, human trafficking, proliferation of weapons of mass destruction, and terrorism.

For instance, it is estimated that five to fifteen percent of all large vessels (that is, 5 000 to 7 500 vessels) break the law each year by discharging waste into the high seas, including 70 to 210 million gallons of illegal oil waste disposal [92]. Such negligence and inconsideration can potentially devastate the marine environment on a local or even global scale. It is also estimated that a third of all fish populations are over-exploited or have collapsed because of illegal fishing [39]. As a result, some of these species face a constant danger of extinction, and over-exploitation is estimated to generate an indirect annual cost of US $50 billion in lost fishing opportunities [111], which accounts for approximately half of the value of the global seafood trade. Moreover, the global economic cost resulting from acts of piracy is estimated to lie between US $7 and US $12 billion per annum [10].

1.2 Informal problem description

Pursuing the goal of effective *Maritime Law Enforcement* (MLE) requires coastal nations to establish suitable monitoring procedures aimed at vessels within their jurisdictions. These procedures are underpinned by a so-called *response selection* process, where, following the detection and evaluation of potentially threatening events involving *vessels of interest* (VOIs) at sea, *MLE resources*, such as high-speed interception boats, military vessels, helicopters, and/or seaplanes, are dispatched by coast guards and related authorities to intercept and investigate these threats. MLE resources are generally either allocated for the purpose of intercepting VOIs at sea (such resources are said to be in an *active* state), or are strategically allocated to certain patrol circuits or bases until needed for future law enforcement purposes (such resources are said to be in an *idle* state). Additionally, MLE resources may temporally be unavailable for law enforcement operations over certain periods of time due to routine maintenance, infrastructure damage, unavailability of crew or depleted autonomy prerequisites. MLE resources which are both idle and assigned to a patrol circuit are said to be on *stand-by*. In this dissertation, the MLE response selection operations considered focus almost exclusively on the management of active MLE resources.

Shortages of law enforcement infrastructure, large jurisdiction coverage areas, high operating costs of MLE resources, the requirement of using complex threat detection and evaluation systems, scarce maritime intelligence and a lack of adequately trained operators are examples of factors contributing to the difficulty of effective MLE by coastal nations, inevitably affecting their overall ability to achieve effective counter-threat performance at sea. A simplified hypothetical MLE scenario, depicting the kind of visual information that an MLE response selection operator may be observing on a human-computer interface in order to assist him in making MLE response selection decisions, is portrayed in Figure 1.2.

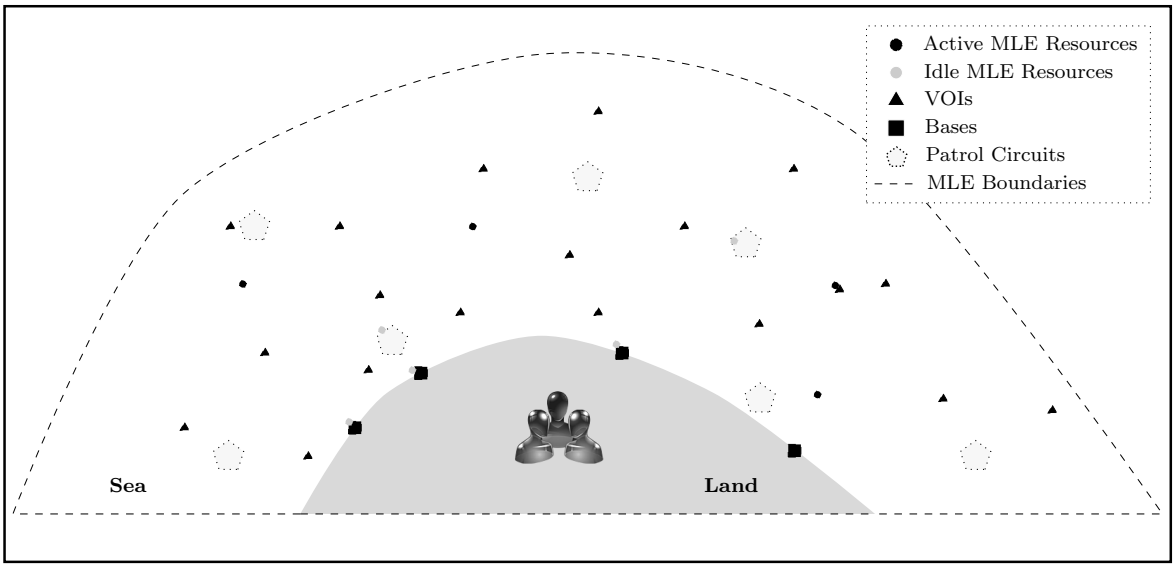


FIGURE 1.2: *Top view of a hypothetical MLE environment.*

MLE operations often comprise complex tasks, typically involving a number of explicitly or implicitly identified subtasks, each with specific resource capability requirements that need to be matched with the capabilities of available MLE resources in order to ensure successful VOI interception [54]. These tasks are stochastically distributed in both time and space, making the

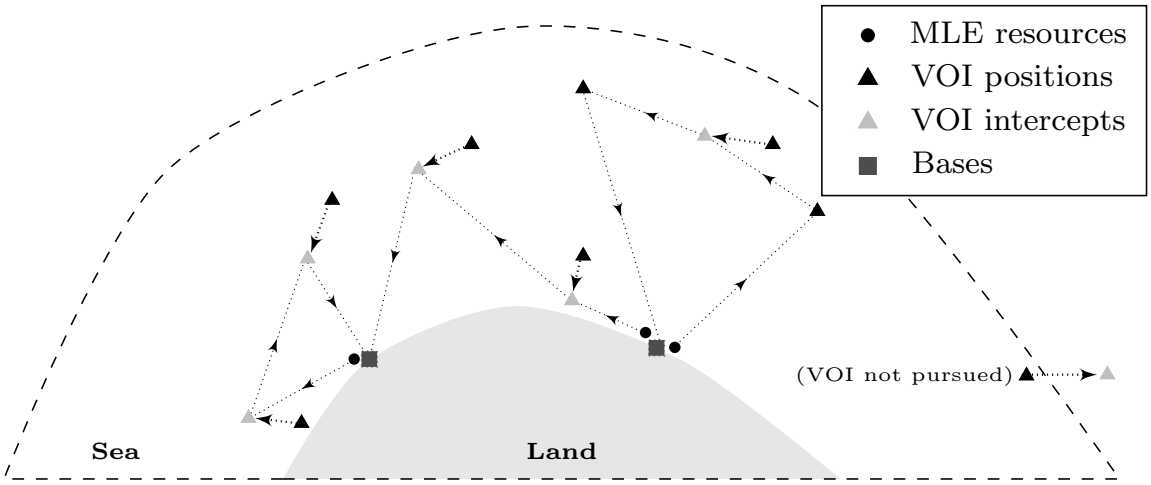
coordination of these MLE resources, which operate in a harsh and unpredictable environment, a very challenging problem. Additionally, the ability to adapt dynamically to changes in the availability of MLE resources and the services they provide is critical for the success of MLE efforts.

The assignment of MLE resources to intercept VOIs requires the formation of so-called *visitation routes*. These are ordered sets containing specific subsets of VOIs that are scheduled to be intercepted and investigated by specific MLE resources over the course of a certain time horizon. This concept is known in the literature as *vehicle routing*. Initial MLE resource deployment is, however, typically carried without full information in respect of the current and future maritime situation. Ancillary information is expected to be gathered from external sources on a continual basis and contribute to the evolution of VOI threat assessment, providing operators with access to actionable information. In order to achieve MLE efficiency, it is therefore required that these decisions be made and coordinated in such a way as to enable the rapid and semi-autonomous re-deployment of MLE resources at various points in time as the sea picture unfolds. This phenomenon is known in the literature as *system dynamism*. Moreover, several input data are not known with certainty, but are rather described by random variables with known or estimated probability distributions. This phenomenon is known in the literature as *system stochasticity*.

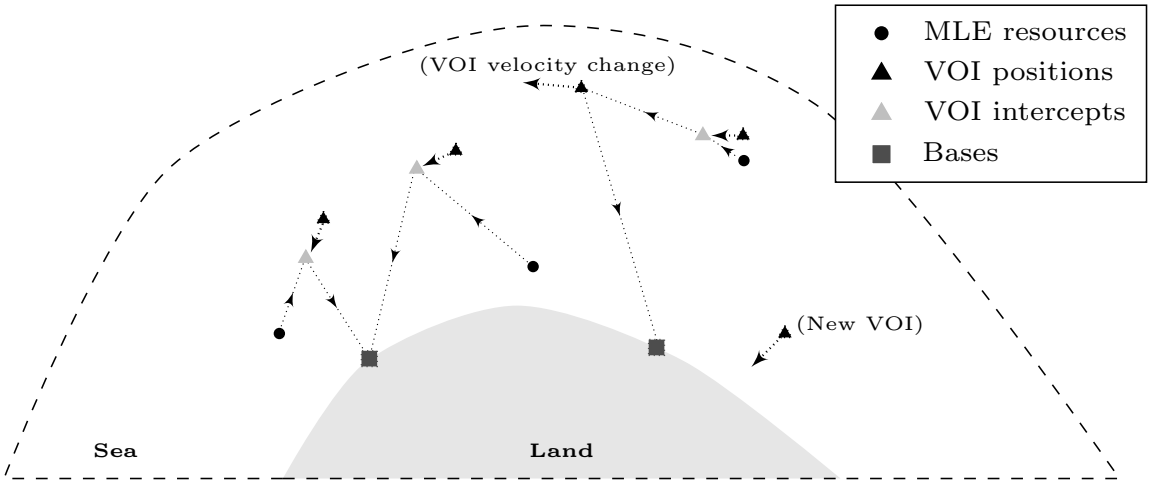
The notions of MLE resource routing and system dynamism mentioned above are elucidated by means of a hypothetical MLE response selection scenario whose evolution over time is illustrated in Figure 1.3. At first, an initial set of MLE resource visitation routes (denoted by directed dotted arcs) is generated, as shown in Figure 1.3(a), in which MLE resources are each allocated to a subset of VOIs. Here, the arrows pointing outwards from VOIs (denoted by black triangles) represent their respective velocity vectors (not to scale), while their estimated interception points are denoted by gray triangles. Later on in this scenario, however, after the MLE resources have only covered parts of their visitation routes, suppose that two new *events* are observed in the system, as shown in Figure 1.3(b). In the first event, one of the VOIs in the north of the jurisdiction area, which was previously immobile, begins to accelerate towards the west. In the second event, a new VOI is detected in the south-east of the jurisdiction area and its threatening nature is evaluated. Given this new information, the current MLE routing solution is re-assessed, and the initial, partially completed visitation routes, are replaced with a new set of more appropriate routes, as shown in Figure 1.3(c).

In order to achieve this level of dynamism, it is necessary to analyse the situation at sea whenever new events are observed and to update the current response for dealing with the potential threats. A semi-autonomous MLE response selection *Decision Support System* (DSS) may be employed to assist human operators in solving the so-called *MLE response selection problem* described above — which encompasses allocation and routing decisions for MLE resources for the purpose of intercepting and investigating VOIs. The purpose of such a DSS is to provide the operators with a set of high-quality response selection alternatives in limited time, particularly when dealing with decisions involving large numbers of VOIs that are subject to a high level of stochasticity with respect to their nature and hence uncertainty. The output of such a DSS may then be used in conjunction with operator judgment to select a single, most preferred alternative from a non-dominated set of solutions, typically in a multiobjective decision space.

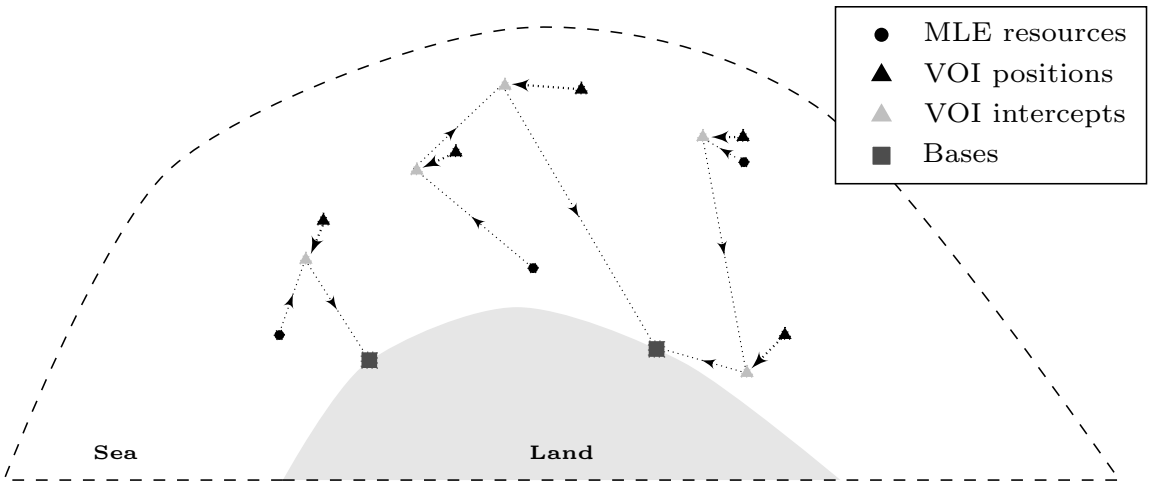
Since each coastal nation has its own values, preferences and perceptions of the desirability of trade-offs between a miriad of objectives when dealing with VOIs, MLE responses following the detection and evaluation of new events at sea typically vary from nation to nation. These responses should, however, be coherent and carried out according to a pre-determined protocol, based on a set of goals and objectives appropriate for the coastal nation in question. A deep understanding of the specific strategic aims and subjective preferences of a coastal nation's MLE



(a) *Initial set of routes allocating MLE resources to subsets of VOIs.*



(b) *Partially completed initial set of visitation routes when the two new events are observed.*



(c) *Set of routes after re-evaluation of the situation, taking into account new information as a result of significant changes in the maritime picture.*

FIGURE 1.3: *Hypothetical example illustrating the notions of routing and system dynamism.*

efforts is therefore necessary in order to identify a suitable set of fundamental objectives for use in the creation of that nation's MLE response strategy.

Furthermore, it is typically the case that the entire MLE response selection process of a coastal nation is not conducted by a centralised operator assisted by a single DSS, but is rather orchestrated by multiple role players, called MLE *decision entities* in this dissertation. These decision entities may perceive the quality of MLE response selection operations in their own, different ways, as they each tend to pursue their own goals and subjective perception of what is deemed important while carrying out MLE operations. In particular, these decision entities may perceive the threatening intensities of VOIs differently, function largely independently from one another, follow their own guidelines or rules of engagement, and utilise their own subsets of MLE resources.

Various MLE resource assignment scenes from around the world are illustrated in Figure 1.4. The high-speed vessel represented in Figure 1.4(a) is one that is used to patrol the waters around Cape Town. Patrolling allows MLE resources to be placed strategically, on expectation, until assigned to investigate one or more VOIs at sea as the need arises. There is, of course, a trade-off involved in having certain MLE resources patrol territorial waters between the costs of operating those MLE resources effectively (instead of being idle at a base) and the expected response times of these MLE resources once becoming active again.

The five vessels aligned in Figure 1.4(b) represent five different types of MLE resources available to the Canadian Coast Guard. A coastal nation typically possesses a fixed number of MLE resources, each belonging to a certain type or class of MLE resource with unique characteristics (such as maximum speed, travel costs per kilometre, level of autonomy at sea and ability to respond to certain types of threats). Two further such characteristics are the set-up time and set-up cost — an MLE resource is required to undergo a careful preparation process (including refueling, briefing of the crew and the preparation of on-board equipment), which takes up a certain amount of time and incurs certain costs in the process, prior to being dispatched on a mission. An example of an MLE resource in the set-up process at a base is illustrated in Figure1.4(c).

Some VOIs may only be detected far out in the jurisdiction area of a coastal nation. In such cases, it may be necessary to dispatch long-range MLE resources to intercept those VOIs. Search-oriented MLE resources, such as seaplanes or unmanned airplanes, may be deployed to visually scout out the precise locations of VOIs that are difficult to track by radars and relay this and other information back to the intercepting vessel commander. Such resources can be seen in Figures 1.4(d) and 1.4(e). Moreover, the threatening nature of a VOI may not always be known *a priori*, in which case these scouting MLE resources — unmanned airplanes in particular due to their stealth, speed, and high-altitude autonomy — may also be utilised to assess the VOI threat situation visually and relay relevant information back to a threat evaluation operator.

Finally, as was discussed in §1.1.2, coastal nations around the world are faced with many different types of threats of differing intensities. Piracy and oil pollution are usually seen as very harmful threats; examples of neutralising such threats are illustrated in Figures 1.4(f) and 1.4(g), respectively. Certain situations may also require an operator to allocate multiple MLE resources to a VOI in order to successfully neutralise it, such as in the scene depicted in Figure 1.4(h), in which two Japanese Coast Guard MLE resources are utilised to neutralise a VOI carrying dangerous Chinese activists.

(a) *Light, high-speed patrol vessel used in waters around Cape Town [22].*

(b) *Various types of MLE resources available to the Canadian Coast Guard [57].*

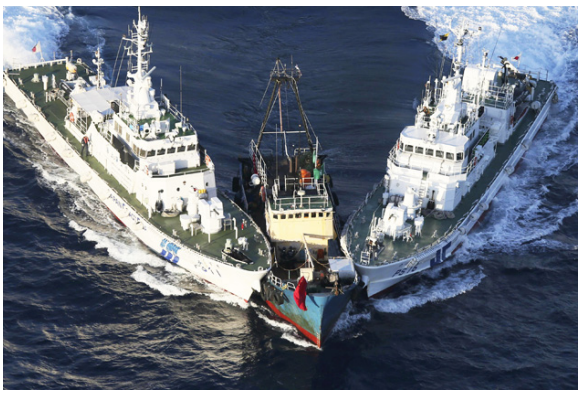
(c) *US coast guards in the process of setting up several MLE resources prior to dispatch [94].*

(d) *Seaplanes are effective scouts when deployed in parallel with long-range MLE resources [94].*

(e) *Unmanned airplanes are popular for acquiring threat evaluation information [70].*

(f) *Interception of a suspicious boat thought to carry ammunition off the coast of Somalia [59].*

(g) *Preventative measures against a potential oil spill situation off the Panama coast [99].*



(h) *Combined assignment of MLE resources onto a high-alert VOI in the East-China sea [63].*

FIGURE 1.4: *Various MLE response selection scenes from around the world.*

1.3 Dissertation aim and scope

The aim in this dissertation is to contribute on an advanced research level to coastal MLE response selection and routing decision infrastructure in the form of the design of a semi-automated DSS capable of assisting human operators in spatio-temporal resource dispatch decision making so that MLE resources may be employed effectively and efficiently. As explained in §1.2, these decisions are made based on input data (which are assumed throughout this dissertation to be complete and accurate) obtained from the threat detection and threat evaluation systems employed with a view to analyse the potentially threatening behaviour of vessels observed at sea at any given time, as well as subjective input data contributed by MLE response selection operators[2]. Automated decision making is therefore not pursued as the aim of this study is to provide a support tool to a human operator.

The multiple objectives component of the proposed DSS will be incorporated in the system in a generic manner, so as to provide users of the system with the freedom of configuring their own, preferred goals. This generic DSS design will be populated with examples of models capable of performing the functions of the various constituent parts of the system, and their workability will be tested and demonstrated by solving the MLE response selection problem in the context of realistic, but simulated, practical scenarios, employing multiple solution search techniques. These examples may, of course, be replaced with more desirable or better performing features, were such a DSS to be implemented in a realistic environment in the future.

Although difficult to measure against a simple MLE response selection human operator in terms of performance, this system may be used in the future to assist such an operator in his decision making process. In particular, the operator may use it as a guideline to validate and/or justify his decisions when the level of uncertainty pertaining to the observed maritime scenario is high and/or if only parts of the problem may be solved with the sole use of his competence and expertise in the field. Overall, usage of this system in a real-world context may reduce operator stress levels typically associated with difficult decisions, while improving the overall performance of MLE operations in an integrated fashion across multiple decision entities.

The research hypothesis of this dissertation is therefore as follows: *Semi-automated decision support based on mathematical modelling principles may be used to assist in making better decisions in an MLE response selection context than those based purely on subjective human operator judgment.*

1.4 Dissertation objectives

The following objectives are pursued in this dissertation:

I To *compile* a literature survey of the theoretical concepts underlying the design of an MLE response selection and routing DSS, by

  (a) *discussing* the nature of decision making procedures in general,

  (b) *presenting* an array of popular stochastic search techniques used to solve complex discrete optimisation problems,

  (c) *elaborating* on the workings of two such techniques in the context of multiobjective optimisation,

[2] This process is better known as a "human in the loop" approach to decision making [49].

(d) *studying* various relevant variants on the vehicle routing problem,

(e) *focusing on* certain procedures designed to cope with the intricacies of dynamic multiobjective optimisation problems, and

(f) *reviewing* key MLE DSSs in existence.

II To *design* a generic MLE response selection framework and overall MLE system architecture.

III To *propose* suitable paradigms of decision entity autonomy in which the MLE response selection framework of Objective II may function in a practical manner.

IV To *derive* mathematical descriptions for each MLE response selection paradigm proposed in Objective III, by

(a) *identifying* the system components required for effective response selection and routing decision support,

(b) *investigating* the functioning of such components and their integration within an MLE response selection environment, and

(c) *incorporating* these components into the formulation of a generic design populated with examples of (replaceable) mathematical models.

V To *educe* an array of non-deterministic mathematical model components capable of managing the risk associated with the potentially adverse impact of stochastic input data on real-time solution development.

VI To *put forward* a number of solution search techniques capable of solving the MLE response selection problem approximately.

VII To *determine* experimentally whether the proposed DSS is capable of producing relatively good solutions in limited computational time, by

(a) *designing* realistic, but hypothetical, MLE response selection and routing scenarios,

(b) repeatedly *solving* the MLE response selection routing problem in the context of the scenarios in (a) using the solution search techniques put forward in Objective VI and different choices of parametric configurations within these solution search techniques,

(c) *comparing* the performances of the solution search techniques in (b) above using a widely accepted performance indicator, and

(d) *analysing* the results obtained, drawing conclusions and commenting on the validity of the research hypothesis.

VIII To *consolidate* the evolution of information into a multi-level, dynamic framework capable of triggering rapid and effective MLE response selection problem instance reinitialisation protocols at various points in time, as required.

IX To *suggest* a number of ideas for possible future work in the context of MLE threat evaluation and MLE resource assignment as enhancements to the work contained in this dissertation.

1.5 Dissertation organisation

This dissertation comprises nine further chapters, following this introductory chapter. In Chapter 2, a literature survey is performed in respect of the theoretical concepts underlying the design of an MLE response selection and routing DSS. A discussion on the philosophy and fundamental concepts behind the underlying decision making process is conducted and an overview of appropriate solution techniques is provided, with a particular emphasis on the use of metaheuristics. The method of simulated annealing and the notion of an evolutionary algorithm are covered in detail, and a general description of three other innovative search techniques is also given. A review of routing problems and dynamic multiobjective problems is finally pursued, and this is followed by a review of MLE semi or fully automated DSSs already in existence.

In Chapter 3, a generic system architecture is proposed for use in coordinating the MLE operations of a coastal nation. Such a system is put forward in order to capture the essence of the process involved in MLE response selection as well as to describe the typical chronological order of MLE events. Moreover, a discussion of three hypothetical paradigms with respect to the autonomy of decision makers using the DSS is conducted in order to capture the roles of the various decision entities typically responsible for solving the MLE response selection problem.

Chapter 4 contains various preliminary concepts, ideas and assumptions underlying the MLE response selection problem. In particular, a description of the modelling approach adopted in respect of mathematically incorporating each of three decision making paradigms is presented.

The focus in Chapter 5 shifts to a system component which aims to accommodate a variety of potential special requests in the system prior to launching the solution search process. This component provides the operator with a range of configuration modules that may be used to implement personal preferences as well as to reduce the computational complexity of the problem using certain *a priori* conditions.

The limitations associated with exclusive deterministic modelling in the context of MLE response selection are unveiled in Chapter 6. The need to manage uncertainty in the form of various mathematical model components dealing with stochastic input data is subsequently motivated. More specifically, this chapter focuses on investigating the extent to which the development of a set of routes in real time is expected to differ from the *a priori* configuration of these routes, as well as to provide the operator with certain modelling tools aimed at minimising or nullifying the risk of solution deterioration caused by stochastic input data.

Four appropriate solution search techniques for solving the MLE response selection routing problem are described in some detail in Chapter 7. Other topics covered in Chapter 7 include an algebraic approach towards calculating the interception point of two moving objects in a two-dimensional plane and a suitable solution encoding scheme adapted specifically for the MLE response selection problem.

In Chapter 8, the optimisation methodologies proposed in the previous chapter are tested in respect of their abilities to yield high-quality solutions to a hypothetical scenario involving the MLE response selection. Methods of filtering and presenting to the decision maker the various trade-off solutions thus uncovered are also provided.

The dynamic nature of MLE response selection operations is elucidated in Chapter 9 by considering the time continuum in which problem instance reinitialisation procedures and solutions progressions are mapped sequentially in a functional and effective manner. A dynamic multi-level framework is proposed for this purpose in which the synchronised use of real-time system memory is combined with three DSS dynamism components to, ultimately, enhance the execu-

tion of the problem instance reinitialisation methodology as disturbances occur in the picture unfolding at sea.

Finally, the dissertation closes with a short conclusion in Chapter 10 in which the work presented in the dissertation is summarised. The chapter also contains an appraisal of the contributions of the dissertation as well as a number of ideas and recommendations with respect to possible future follow-up work.

CHAPTER 2

Literature Review

Contents

This chapter contains a review of the literature on topics related to the design of a DSS for assisting operators in their complex MLE response selection decisions. The purpose of the chapter is to provide the reader with the necessary background in order to facilitate an understanding of the material presented in the remainder of this dissertation.

In §2.1, a discussion is conducted on the philosophy and fundamental concepts behind decision making, while an overview of solution techniques for optimisation problems is provided in §2.2, with particular emphasis on the use of metaheuristics. The methods of simulated annealing and evolutionary algorithms are then presented in more detail in §2.3 and §2.4, respectively, and this is followed by a general description of three other innovative metaheuristic search techniques in §2.5. A review of routing problems with emphasis on dynamic and stochastic routing problems may be found in §2.6. In §2.7, a closer look is taken at various approaches used in the literature toward solving multiobjective dynamic optimisation problems, while the focus turns in §2.8 to a review of MLE semi or fully automated DSSs in existence. The chapter finally closes with a brief summary in §2.9.

2.1 Decision theory

The etymology of the term *decision* derives from the Latin word *decidere*, which means *to decide* or to *determine*. This word consists of the words *de*, which means "off", and *caedere*, which means "to cut" [30]. According to this description, *decision making* is the process whereby plausible alternatives with respect to the course of actions to be taken when exposed to a

certain situation are rejected from a finite set of choices (*i.e.* cut-off) until all that remains is a most desirable alternative to be executed as a response to the situation. This definition is particularly applicable in the field of operations research, in which the feasible *decision space* of a problem is traditionally analysed, or "searched", and alternative solutions to the problem are rejected according to a certain sequence of rules until a single (*i.e.* most preferred) alternative (or set of alternatives) remains. Decision making is a critical component in almost any situation. Although most decisions take place on the subconscious level, the ability to make high-quality choices under uncertainty may usually only be realisable with the use of some complex analytical process.

2.1.1 Elementary concepts of decision making

In decision making, a *goal* may be thought of as a purpose towards which an endeavor is directed, while an *objective* may be thought of as a specific result or results that a person or system aims to achieve within a certain time frame and with available resources. Whereas goals do not necessarily have to be measurable or tangible, objectives are expected to be so [46]. The choice of alternatives in a decision making situation should be measurable in terms of evaluative *criteria*. A criterion is a principle or standard which provides a means to compare a set of alternatives on the basis of some particular aspect that is relevant or meaningful to the decision maker(s) [27]; they help define, on the most fundamental level, what "best" really means. Objectives are measured on suitable *attribute scales*, which provide a means to evaluate the quality of each alternative with respect to every criterion relevant to the decision problem.

In order to identify these criteria, it is critical to first determine and assess the basic *values* of the decision maker with respect to the situation at hand (*i.e.* what really matters to him) [27]. Values underlie decision making in a fundamental manner; if a decision maker does not care about the criteria, there would be no reason for him to make a decision in the first place, as he would be completely indifferent with respect to assessing the desirability of alternative solutions. Objectives are therefore necessary in order to establish a direct relationship with one or more particular criteria in terms that relate directly to the basic values of the decision maker. In the field of operations research, analysts and consultants unfortunately often neglect such values, instead tending to solve a decision making problem with respect to what *they* think is deemed important, and thereby disregarding certain critical aspects associated with the psychological needs and preferences of their clients[1] [61].

In general, the goal in any decision making process is to seek the best alternative by considering all decision-impacting criteria simultaneously whilst acknowledging decision space delimitations. Incorporating these criteria, as well as the values of the decision maker, in the process is achieved by configuring the objective(s) accordingly. As hinted at previously, every decision making situation is relevant within a specific context, which calls for specific objectives to be considered. A criterion may either be advantageous or disadvantageous with respect to the preferences of the decision makers; alternatives simultaneously minimising or maximising each criterion, respectively, are therefore sought.

2.1.2 On the decision structuring process

According to Clemen & Reilly [27], modelling a decision making problem consists of three fundamental steps: (1) identifying the values of the decision maker with respect to the problem

[1] This discrepancy has given rise to the field of *behavioural operational research* over the last decade.

at hand and defining all objectives so as to correctly integrate the decision model, (2) structuring the elements of the decision making problem into a logical framework, and (3) defining and fully refining all the elements of the model.

Furthermore, the set of objectives should include all relevant aspects of the underlying decision yet be as small as possible (so as to avoid unnecessary computational complexity). The set of objectives should therefore not contain redundant elements and should be decomposable[2]. Finally, the objectives should be clearly distinguishable and the relevant attribute scales should provide a simple way of measuring the performance of alternatives with respect to the objectives.

Generally speaking, objectives may either be categorised as *fundamental* or *mean* [27]. Fundamental objectives are the basis on which various alternatives may be measured (the type of objectives discussed in the previous section). Mean objectives, on the other hand, are of a purely intermediate nature, and are used to help achieve other objectives. While fundamental objectives are organised into hierarchies, which are crucial for the development of a multiobjective decision model, mean objectives are usually organised in the form of networks (similar to a brainstorming configuration).

Influence diagrams and *decision trees* are examples of techniques often used to structure the decision making process into a logical framework [27]. An influence diagram captures the decision maker's current state of knowledge by providing a simple graphical representation of the decision situation. In such a diagram, the decision process is represented as various shapes, interlinked by directed arcs in specific ways so as to reflect the relationship among the decision components in a relevant, sequential order. Although appropriate for displaying a decision's basic structure, influence diagrams tend to hide details. A decision tree, on the other hand, expands in size and level of detail in a time sequence as the decision process evolves. Such a tree represents all possible paths that the decision maker might follow throughout the decision making process, depicting all alternative choices, as well as consequences resulting from uncertain events. The alternatives, represented by branches leaving a decision node, must be such that only one option can be chosen, while the branches leaving uncertain event nodes must correspond to a set of mutually exclusive and collectively exhaustive outcomes (that is, at most one outcome may occur from a finite set of outcomes with specific probabilities, and at least one of these outcomes has to occur). Overall, both influence diagrams and decision trees have their own advantages for structuring the decision making process. Their combined use in parallel during the decision making process, however, provides a complete model of the decision framework. One may therefore think of them as complementary rather than competitors in structuring a decision making process.

2.1.3 Decision making with multiple conflicting objectives

Multiobjective decision making problems occur in most disciplines. Despite the considerable variety of techniques that have been developed in the field of operations research since the 1950s, solving such problems has presented a non-trivial challenge to researchers. Indeed, the earliest theoretical work on multiobjective problems dates back to 1895, when the mathematicians Cantor and Hausdorff laid the foundations of infinitely dimensional ordered spaces [72]. Cantor also introduced equivalence classes and utility functions a few years later [29], while Hausdorff published the first example of a complete ordering set [29].

In most decision making situations, the decision maker is required to consider more than one

[2] A set of objectives is *decomposable* if the decision maker is able to think about each objective easily without having to consider the others simultaneously.

criterion simultaneously in order to build a complete, accurate decision making model. These criteria, however, typically conflict with one another as informed by the values of the decision maker. Objectives are said to be *conflicting* if trading an alternative with a higher achievement measure in terms of a certain criterion comes at a cost (*i.e.* a decrease in the levels of achievement of some of the other criteria for that alternative). A crucial problem in multiobjective decision making therefore lies in analysing how to best perform trade-offs between the values projected by these conflicting objectives.

There exist multiple techniques for determining the quality of an alternative in a decision making process [27]. An appropriate technique should be selected based on factors such as the complexity of the problem, the nature of the decision maker, the time frame for solving the problem or the minimum quality level of the best alternative. The *additive preference model*, for example, is a popular method in the literature [43, 117], in which the decision maker is required to construct a so-called *additive utility function* for comparing the attribute levels of available alternatives, as well as the fundamental objectives, in terms of their relative importance, using weight ratios. This decision making model approach is, however, incomplete, as it ignores certain fundamental characteristics of choice theory amongst multiatribute alternatives [27]. In order to resolve this discrepancy, slightly more complex methods such as *multiattribute utility models* were designed [104]. Here, the decision maker considers attribute interactions by establishing sub-functions for all pairwise combinations of individual utility functions, in contrast to utilising an additive combination of preferences for individual attributes.

In general, the approach adopted toward solving a multiobjective decision making problem may be classified into two paradigms [27, 81]. The first involves combining the individual objective functions into a single composite function, or to move all except one objective to the set of constraints. This is typically the case in utility theory and weighted sum methods. The additive utility function and multiattribute utility models mentioned above, are examples of decision making techniques in this paradigm. Optimisation methods in this paradigm therefore return a single "optimal" solution to a multiobjective decision making problem. This procedure of handling multiobjective optimisation problems is a relatively simple one. Due to the subjective nature of the decision making problem in terms that depend purely on the decision maker's preferences, however, it may often be very difficult to determine all the necessary utility functions and weight ratio parameters accurately, particularly when the number of attributes to be considered is large. Perhaps more importantly, parts of the front are inaccessible when adopting fixed weights in the case of non-convex problems [42]. This optimisation procedure may be enhanced to some extent by considering multiple *a priori* weight vectors; this is particularly useful in cases where choosing a single appropriate set of weight ratio parameters is not obvious. In most non-linear multiobjective problems, however, it has been shown that a uniformly distributed set of weight vectors need not necessarily result in a uniformly distributed set of Pareto-optimal solutions [42]. Once again, since the nature of this mapping is not usually known, it is often difficult to select weight vectors which are expected to result in a Pareto-optimal solution located within some desired region of the objective space.

Alternatively, the second paradigm, referred to as *multiobjective optimisation*, aims to enumerate and filter the set of all alternatives into a suggested decision set in such a way that the decision maker is indifferent between any two alternatives within the set and so that there exists no alternative outside the set which is preferred to any alternatives within the set. Due to the conflicting nature of the objectives, no single solution typically exists that minimises or maximises all objectives simultaneously. In other words, each alternative in the suggested decision set yields objective achievement measures at an acceptable level without being outperformed by any other alternatives [81]. Decision making techniques within this paradigm therefore aim to

identify a set of alternative solutions which represent acceptable inter-attribute compromises. This approach is elucidated in the following section. One of the principal advantages of multiobjective optimisation is that the relative importance of the objectives can be decided *a posteriori* with the Pareto front on hand.

2.1.4 The notions of dominance, Pareto optimality and the Pareto front

In contrast to single objective optimisation problems, in which a single, optimal solution is sought, multiobjective problems induce a set of alternative solutions in the decision space of the problem which, when evaluated, produce vectors whose components represent trade-offs in the multidimensional *objective space* of the problem. As described in the previous section, the objectives being optimised in a multiobjective optimisation problem usually always conflict with one another, placing a partial, rather than total, ordering on the search space [29].

Consider a multiobjective optimisation model consisting of n decision variables $x_1, \ldots, x_n$ with respective restricting lower and upper bound values $(x_1^{(L)}, x_1^{(U)}), \ldots, (x_n^{(L)}, x_n^{(U)})$, m constraints with standardised evaluative functions $g_1, \ldots, g_m$, k objectives with evaluative functions $f_1, \ldots, f_k$ mapping the n-dimensional vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ in *decision space* (or *solution space*) to the k-dimensional vector $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$ in *objective space*. This mapping may or may not be surjective, depending on the evaluative functions and constraint configurations [42]. The multiobjective optimisation problem in its general form may be stated as:

$$\begin{aligned} \text{Minimise/Maximise} \quad & f_o(\boldsymbol{x}), && o \in 1, 2, \ldots, k, \\ \text{subject to} \quad & g_c(\boldsymbol{x}) \leq 0, && c \in 1, 2, \ldots, m, \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, && i \in 1, 2, \ldots, n. \end{aligned}$$

Let $\Psi \subseteq \Re^n$ be a compact set representing the feasible decision space of the maximisation problem described above, and let $\Phi_i \subseteq \Re$ represent the feasible objective space with respect to the i^{th} objective function; *i.e.* $f_i : \Psi \to \Phi_i$. Then, $\boldsymbol{x}^* \in \Psi$ is a global maximum with respect to f_i if and only if $f_i(\boldsymbol{x}^*) \geq f_i(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Psi$.

The notion of a global optimum for the i^{th} objective function may be extended to the full multiobjective optimisation problem in the following way. A decision vector $\boldsymbol{x} \in \Psi$ is said to *dominate* a decision vector $\boldsymbol{y} \in \Psi$, denoted here by $\boldsymbol{x} \succ \boldsymbol{y}$, if $f_i(\boldsymbol{x}) \geq f_i(\boldsymbol{y})$ for all $i \in \{1, \ldots, k\}$ and if there exists at least one $i^* \in \{1, \ldots, k\}$ such that $f_{i^*}(\boldsymbol{x}) > f_{i^*}(\boldsymbol{y})$. It follows that any two candidate solutions to the mulitobjective problem described above are related to each other in two possible ways only: (1) either one dominates the other, or (2) neither one is dominated by the other. Moreover, $\boldsymbol{x}$ is said to be non-dominated with respect to the set $\Psi' \subseteq \Psi$ if there exist no vectors $\boldsymbol{y} \in \Psi'$ such that $\boldsymbol{y} \succ \boldsymbol{x}$. Finally, a candidate solution $\boldsymbol{x}$ is said to be *Pareto optimal* if it is non-dominated with respect to the entire decision space Ψ.

According to the above definition, Pareto optimal solution vectors therefore represent trade-off solutions which, when evaluated, produce vectors whose performance measure in one dimension cannot be improved without detrimentally affecting at least some subset of the other $k - 1$ dimensions. The *Pareto optimal set* $\mathcal{P}$ is, henceforth, defined as the set of candidate solutions containing all Pareto optimal solutions in Ψ. That is,

$$\mathcal{P} = \{\boldsymbol{x} \in \Psi \mid \nexists\, \boldsymbol{y} \in \Psi \text{ such that } \boldsymbol{y} \succ \boldsymbol{x}\}.$$

Pareto optimal solutions may have no obvious apparent relationship besides their membership of the Pareto optimal set; such solutions are purely classified as such on the basis of their values

in objective space. These values produce a set of objective function vectors, known as the *Pareto front* $\mathcal{F}$, whose corresponding decision vectors are elements of $\mathcal{P}$. That is,

$$\mathcal{F} = \{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{P}\}.$$

A graphical illustration of certain concepts discussed above is provided in Figure 2.1 for a multiobjective problem for which $n = 3$, $k = 2$ and $m = 6$. Here, elements from the cubic feasible solution space $\Psi \subset \Re^3$ (delimited by the dashed planes), are mapped to the pentagonal objective space $\Phi \subset \Re^2$. The Pareto front is denoted by the bold curve in objective space. Moreover, all objectives are here assumed to be computable functions available in closed form which have to be maximised.

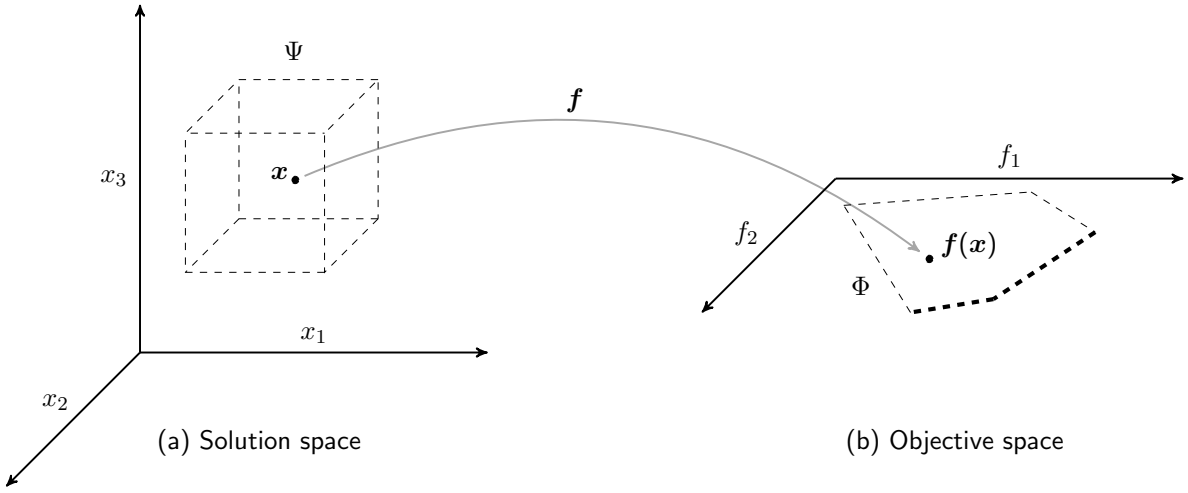


FIGURE 2.1: *Graphical example of the primary features of a multiobjective problem with $n = 3$ decision variables constrained by $m = 6$ constraints, $k = 2$ and objective functions f_1, f_2 which are to be maximised. The Pareto front is denoted by the bold curve in objective space.*

2.1.5 Multiperson decision making

In most decision making situations, the outcome of the decision directly or indirectly affects more than just the decision maker; although he might be the only one responsible for ultimately making the decision, regardless of the impact of the decision outcome on other individuals. In certain cases, however, the decision maker may wish to analyse how the outcome of a particular decision affects a group of individuals, or he may consider individual preferences as part of the decision making process. Alternatively, several decision makers may carry out the decision making process together, with the aim of reaching some form of consensus or agreement. Decisions involving the contribution of multiple members or parties in a decision making process are defined in the literature as *multiperson* decision making problems [65].

Examples of two well-known techniques (from opposite ends of the methodological spectrum) for solving multiperson decision making problems are *majority voting* and *consensus attainment*. Whereas in majority voting, agreement by a set majority of members is sufficient to reach a final decision, most or all members must reach a certain level of mutual agreement in order for the decision process to move forward in a consensus attainment process.

Consensus attainment therefore deals with the cooperative process of obtaining the maximum degree of agreement between a number of decision makers with respect to selecting and sup-

porting a solution from a given set of alternatives, for the greater good of the situation at hand. In practice, the consensus process is typically a dynamic and iterative group discussion process, usually coordinated by a moderator, responsible for helping the members to bring their opinions closer to one another [65]. During each step of the process, the moderator keeps track of the actual level of consensus between the decision makers, typically by means of some pre-determined consensus measure establishing the distance from the current consensus level to the ideal state of consensus (*i.e.* full and unanimous agreement of all members with respect to a specific alternative). If the current consensus level is deemed unacceptable, that is, if it is lower than some pre-determined threshold level, indicating that there exists a considerable degree of discrepancy between the decision makers' opinions, then the moderator asks the members to discuss their opinions further in an effort to bring them closer to consensus. The consensus attainment decision making process is also sometimes able to function without the use of a moderator, whose inclusion in the process may be too time-consuming, but is rather controlled automatically by the group, receiving input from the various decision makers, assessing the consensus level at each step of the process, and providing feedback (output) on the current state and progress of the process back to the decision makers.

According to Bressen [14], while majority voting typically leads to a (technically) much simpler and faster decision making process than does consensus attainment, consensus attainment offers three major advantages over majority voting. First, consensus attainment leads to more effective implementation (when all decision makers' ideas and concerns are accounted for, they are more likely to participate actively towards making something happen). In majority voting, on the other hand, members with a minority position are usually overruled and are naturally unwilling to participate in the decision process with much eagerness. Secondly, consensus attainment builds connections amongst decision makers (using consensus attainment suggests that some time is taken to achieve a certain level of agreement amongst the members on how to proceed with the decision making process before actually moving forward). Majority voting, on the other hand, creates winners and losers, which has a dividing effect amongst the decision makers [14]. Lastly, consensus attainment usually leads to higher quality decisions (integrating the wisdom and experience of all members into the decision making process typically generates better and smarter decisions than majority voting does, particularly when the decision makers are assumed to be rational).

2.2 Optimisation techniques

In this section, an overview is provided of general search techniques aimed at finding exact or approximate solutions to optimisation problems. This is followed by a more elaborate discussion on the class of stochastic search techniques, members of which are later implemented in this dissertation. There does, however, not seem to be full consensus on a standard classification for optimisation search methods in the literature.

According to Coello *et al.* [29], one way to differentiating between different types of solution search and optimisation techniques is to classify them as *enumerative*, *deterministic* or *stochastic*. Such techniques may further be classified as *exact*, *heuristic* or *metaheuristic* [153]. Examples of popular search techniques in these categories are listed in Figure 2.2. Other classifications of optimisation techniques may also include partitions according to traditional *versus* non-traditional, local *versus* global or sequential *versus* parallel techniques.

Full enumeration techniques evaluate every possible solution in the solution space of the optimisation problem, either explicitly or implicitly. If the problem complexity is relatively low

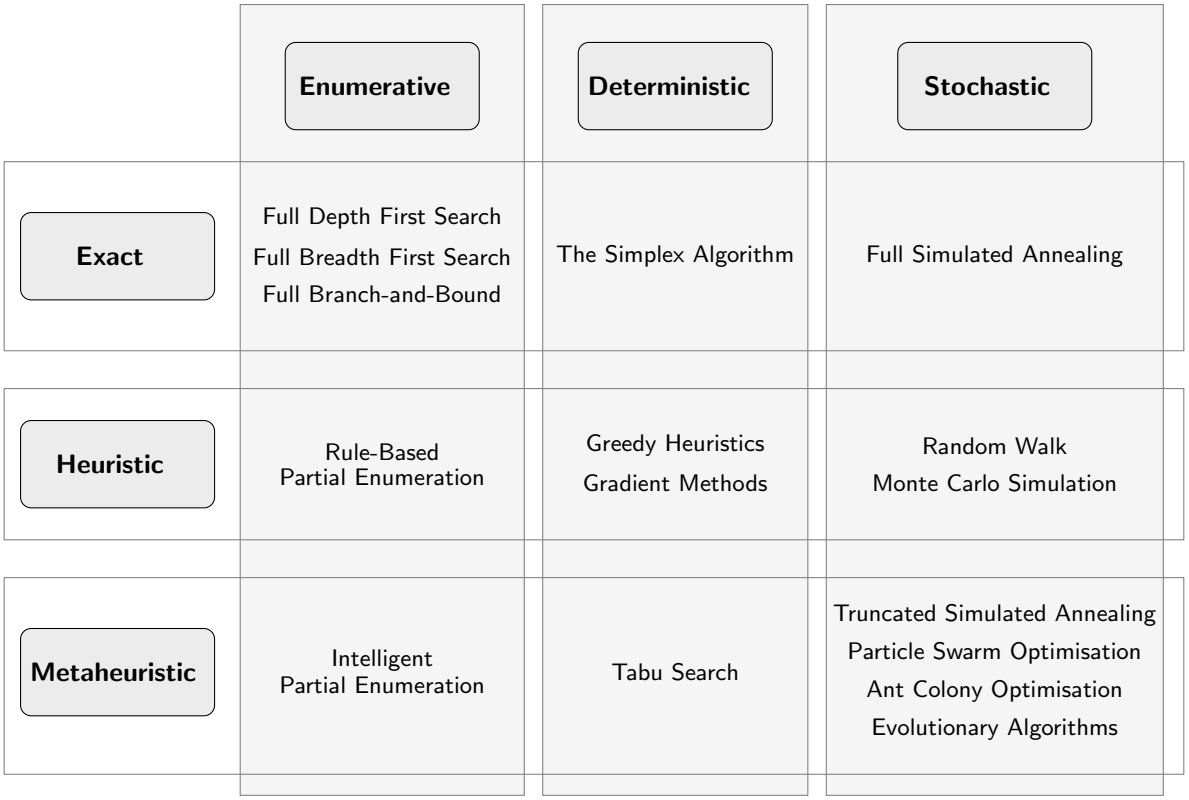


FIGURE 2.2: *Global search categories and examples of techniques.*

and the search can be performed within a reasonable amount of time, then these techniques are guaranteed to find a global optimum. Adopting this kind of approach when solving an optimisation problem is typically extremely straightforward and easy to apply. Enumeration approaches may, however, be entirely insufficient when dealing with large optimisation problems.

Deterministic search techniques overcome the problem of a large search space, by incorporating the notion of *problem domain knowledge* to reduce the size of the search space. Here, the aim is to implement some means of guiding or restricting the search space in order to find good solutions within an acceptable time. Although proven to be successful methods for solving a wide variety of problems, the complex features associated with many real-world optimisation problems, such as multimodality[3], highly dimensionality, discontinuous and/or NP-completeness, often cause deterministic methods to be ineffective for solving problems of this kind[4] over large search spaces, due to their dependence on problem domain knowledge for guiding the search process [29].

Stochastic search techniques were developed as an alternative for solving such irregular problems. Here, the subsequent state of the search process is induced probabilistically (*i.e.* randomly) instead of deterministically. These techniques are based on probabilistic sampling of a set of possible solutions and, throughout the search, maintain some form of record of good solutions found. A predetermined function assigning performance values to candidate/partial solutions as well as a mapping mechanism linking the problem domain to the algorithm is required. Given

[3]A problem is said to be multimodal if it contains an objective function with more than one optimum (*i.e.* one or more global optimum in addition to one or more local optimum of inferior quality) [162].

[4]Problems exhibiting one or more of these characteristics are sometimes called *irregular* [29].

the level of complexity of some irregular multiobjective problems, stochastic methods are often able to yield satisfactory solutions to such problems for which the search space is not chaotic[5].

Metaheuristics have recently become very popular for solving multiobjective optimisation problems. A metaheuristic is a state-of-the-art search technique which may be defined as a higher-level procedure designed to strategically guide one or more lower-level procedures or heuristics to search for feasible solutions in spaces where the search task is hard [31]. Metaheuristics are characterised by their typically approximate, stochastic and non problem-specific nature. According to Suman and Kumar [138], the increasing acceptance of these search techniques is a result of their ability to: (1) find multiple candidate solutions in a single run, (2) function without the use of derivatives, (3) converge with great speed and accuracy towards Pareto optimal solutions, (4) accommodate both continuous and combinatorial optimisation problems with relative ease and (5) be less affected by the shape or continuity of the Pareto front.

One way of classifying metaheuristics is to refer to their search strategies, which, according to Blum and Roli [7], is either *trajectory-based* or *population-based*. These categories are typically characterised by the number of candidate solutions generated during every iteration of the search process. Trajectory-based metaheuristics start with a single initial candidate solution and, at every iteration, replace the current solution by a different, single candidate solution in its neighbourhood (examples of such metaheuristics include simulated annealing [138], tabu search [48] and variable neighbourhood search [123]). Population-based metaheuristics, on the other hand, start with an initial population of multiple candidate solutions, which are enhanced through an iterative process by replacing part of the population with carefully selected new solutions (examples of such metaheuristics include genetic algorithms [29], ant colony optimisation [47] and particle swarm optimisation [114]). Trajectory-based approaches are usually able to find locally optimal solutions quickly, and are thus often referred to as exploitation-oriented methods. Population-based approaches, on the other hand, strongly promote diversification within the search space, and are thus often referred to as exploration-oriented methods. Additionally, population-based approaches often incorporate a learning component.

Other classes of stochastic search techniques include *hybrid* metaheuristics, *parallel* metaheuristics and *hyperheuristics*. A hybrid metaheuristic typically combines several optimisation approaches (such as other metaheuristics, artificial intelligence, or mathematical programming techniques) with a standard metaheuristic. These approaches run concurrently with one another, and exchange information, in order to guide the search process. Parallel metaheuristics, on the other hand, use the techniques of parallel programming to implement multiple metaheuristic searches in parallel so as to guide the search process more effectively. Hyperheuristics were initially introduced to devise new algorithms for solving problems by combining known search techniques in ways that allow each of them to compensate, to some extent, for the weaknesses of others [17]. Their goal is to be generally applicable to a large range of problems by performing well in terms of computational speed, solution quality, repeatability and favourable worst-case behaviours, in cases where standard metaheuristic techniques fail to perform well on some of these counts.

In the following three sections, five metaheuristics are presented in varying amounts of detail. For reasons that will later become apparent, simulated annealing and evolutionary algorithm techniques are discussed in more detail and are reviewed in the context of both single- and multiobjective optimisation problems, while the remaining algorithms are only briefly discussed in the context of single-objective optimisation problems.

[5]A problem is said to be *chaotic* if small differences in initial conditions yield widely diverging outcomes in the long run, rendering long-term predictions near-impossible [149].

2.3 The method of simulated annealing

The method of *simulated annealing* is a metaheuristic that exploits the deep and useful connection between *statistical mechanics* and large combinatorial optimisation problems. It was initially developed for use in the context of highly non-linear optimisation problems [18, 78]. Statistical mechanics may be described as the central discipline of condensed matter physics, embodying methods for analysing aggregate properties of the large number of molecules to be found in samples of liquid or solid matter.

The notion of simulating the evolution of the state of a system towards its *thermodynamic equilibrium* at a certain temperature dates back to the work of Metropolis *et al.* [100] in 1953. In their paper, they present an algorithm (now called the *Metropolis algorithm*) which simulates the behaviour of a collection of molecules in thermodynamic equilibrium at a given temperature. Each iteration of this process consists of moving an atom by a small random displacement, consequently generating a certain change Δ_E in the energy state of the system. If the move results in a decrease in the energy of the system, it is accepted. If the move results in an increase in the energy of the system, it is accepted with a probability $\exp(\frac{-\Delta_E}{T})$, where T represents the current temperature of the system. This occasional increase in objective function value prevents the system from becoming trapped at a local optimum. Repeating this process generates a sequence of configurations which may be modelled as a Markov chain, in which a certain state of the chain represents a state of energy corresponding to its respective system configuration, eventually causing the system to reach a thermodynamic equilibrium[6]. Once a thermodynamic equilibrium at a particular temperature is reached, the temperature is lowered according to a cooling schedule and a new Markov chain of energy states is configured for that new temperature.

2.3.1 The notion of annealing

A fundamental question in statistical mechanics, evolves around the behaviour of a system as the temperature decreases (for example, whether the molecules remain fluid or solidify). From a practical point of view, low temperature is not a sufficient condition for seeking ground states of matter such as crystalline solids [78]. Experiments that determine such states are conducted according to the process of *annealing*.

Annealing involves the heating and cooling of a material to alter its physical properties as a result of the changes in its internal structure [71]. It is initiated by heating a material to a certain temperature in order to convey a certain level of thermal energy to it. The material is then cooled in stages according to a specific schedule. During this process, the temperature of the material is carefully controlled so that the material spends a significant amount of time close to its freezing point. In doing so, the particles of a solid first randomly rearrange themselves once a liquid stage is reached during the melting phase, and then form a crystalised solid state as the temperature is slowly lowered during the cooling phase. The annealing process is terminated when the system reaches a *solidified* state.

At high temperatures, a material in liquid form allows the molecules to rearrange themselves with more freedom, hence increasing the potential combination of inter-molecular *moves*. Low temperatures, on the other hand, cause the molecules to become more confined due to high energy costs of movement, therefore only allowing a limited amount of possible moves to be conducted in the system. Every time a molecule is moved while a system is in equilibrium,

[6]This research later on allowed for the population of the so-called *Boltzman distribution* of the energy states at a specific temperature to be extracted [48].

a certain amount of energy is released, causing the energy level of the system to change in accordance with the motion of that molecule. A sudden change in the energy state of a system is known as a *perturbation*.

In solidified form, the material is uniformly dense and contains almost no structural flaws. In order for the crystal lattices to form in a suitably organised manner, however, the energy level released in the system during the cooling process has to be as small as possible as it approaches this state. Quickly lowering the temperature, known as *quenching*, causes the material to be out of equilibrium as it reaches the state, resulting in low crystalline order or locally optimal structure which ultimately causes defects in the formation of the crystals [78].

2.3.2 Algorithm outline

The annealing process discussed in the previous section exhibits analogies with the solution of combinatorial optimisation problems. Here, a control parameter mimicking the role of the temperature is introduced in the optimisation process. This parameter must have the same effect as the temperature of the physical system in statistical mechanics, namely conditioning the accessibility of energy states (*i.e.* objective function achievement measures), and lead towards a globally optimal state of the system (*i.e.* a globally optimal solution), provided that it is lowered in a carefully controlled manner. The final solution obtained is analogous to a solidified form of the system.

Under the Metropolis acceptance rule, the role of the temperature in simulated annealing is now more clear. Assuming a minimisation problem, $\exp(\frac{-\Delta_E}{T}) \to 1$ as $T \to \infty$, suggesting that many solutions are accepted while the temperature is high[7]. The algorithm therefore performs similarly to a simple random walk search during these early stages, which greatly favours exploration of the search space. On the other hand, the probability of accepting non-improving moves decreases as the temperature decreases, suggesting that solutions degrading the objective function are less likely to be accepted as the next configuration state in the system, but nevertheless giving the system a small chance to be perturbed out of a local optimum. Non-improving moves are accepted according to a so-called *probability acceptance function*, which may or may not be the Metropolis acceptance rule.

Lowering the temperature is controlled according to a so-called *cooling schedule*. Careful consideration must be given to the nature of the cooling schedule; lowering the temperature too quickly lowers the chances of accepting solutions too quickly which may cause large parts of the solution space to remain unexplored, while lowering the temperature too slowly may result in the consideration of many redundant solutions which do not lead to a high-quality non-dominated front in objective space.

Therefore, as the temperature is reduced, only perturbations leading to small decreases in energy are accepted, so that the search is limited to a smaller subset of the solution space (*i.e* so that exploitation of good regions of the solution space occurs) as the system hopefully settles on a global minimum. The analogy of simulated annealing therefore evolves around the assumption that if the move probability decreases slowly enough, a global optimum may be found. If quenching occurs, however, the search is more likely to lead to a local minimum.

A flowchart of the process of simulated annealing is presented in Figure 2.3. Here, the outer loop controls the temperature schedule, while the inner loop attempts to reach a thermodynamic equilibrium according to the Metropolis algorithm. In addition, a pseudo-code description of the basic working of the method of simulated annealing is given in Algorithm 2.1. In the outlined

[7] In a maximisation problem, moves are accepted with probability 1 if they *increase* the energy of the system.

simulated annealing algorithm, a *move* refers to a modification made to the current solution. The algorithm keeps track of the best solution found thus far during the search, called the *incumbent solution*. Once the system reaches a solidified state, the incumbent solution is reported as an approximate solution to the optimisation problem.

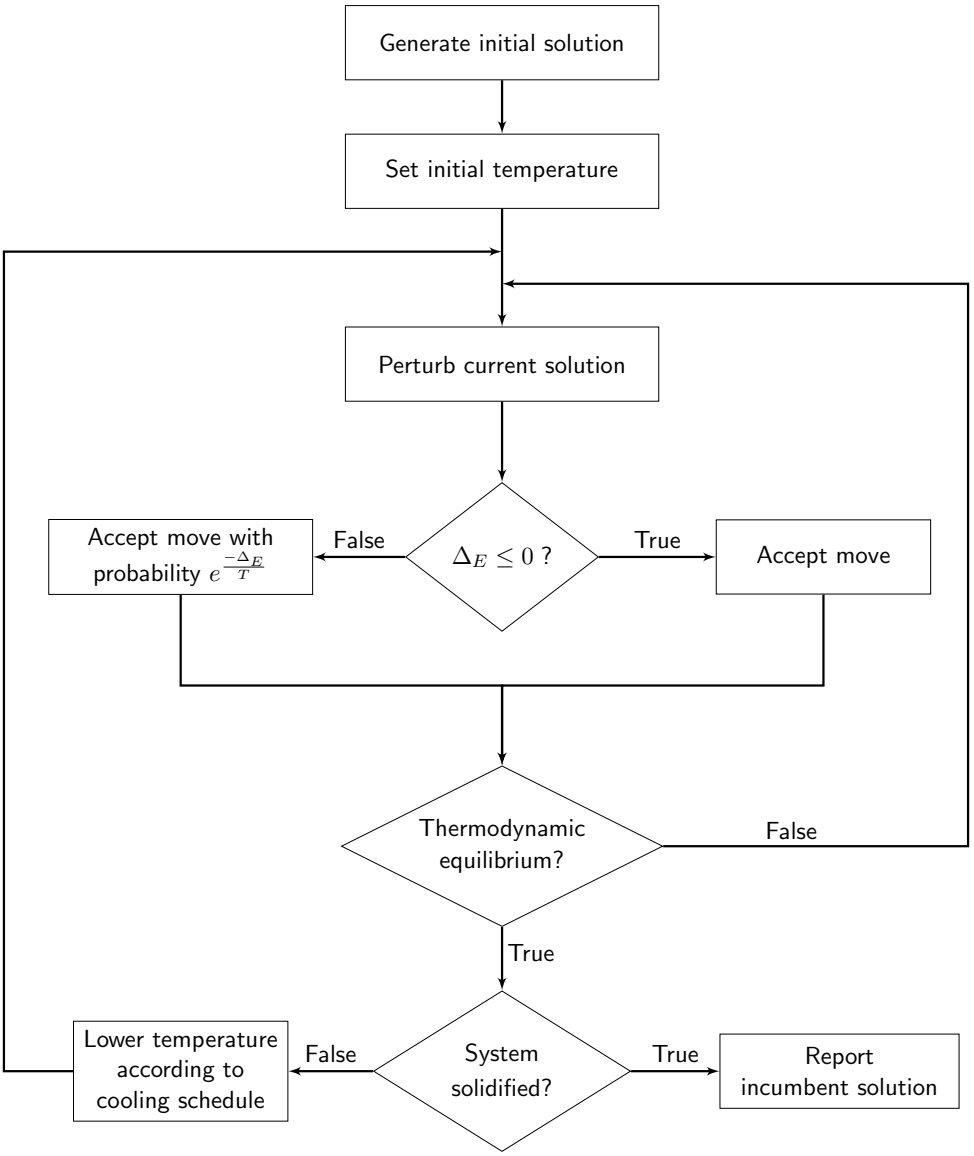


FIGURE 2.3: *Flow of events in the simulated annealing algorithm for a minimisation problem.*

2.3.3 Multiobjective simulated annealing

> *It is possible to compare the relative quality of two vector solutions with the dominance relation, but note that it gives, essentially only three values of quality — better, worse, equal — in contrast to the energy difference in uni-objective problems which usually give a continuum.* — K. Smith [130]

The method of simmulated annealing may also be used to solve multiobjective problems in which the decision maker is eventually presented with an acceptable set of non-dominated solutions.

Algorithm 2.1: Simulated annealing algorithm outline (for a minimisation problem)

```
Generate initial feasible solution
while System not solidified (i.e. T > 0) do
    while Thermodynamic equilibrium not reached do
        Generate neighbouring solution (i.e. perturb current solution)
        Evaluate the change in energy Δ_E resulting from the perturbation
        if Δ_E ≤ 0 then
            Accept the move and make the neighbouring solution the current solution
        else
            Accept the move with probability e^(−Δ_E / T)
    Lower temperature according to the cooling schedule
Report incumbent solution
```

Just like in single-objective optimisation problems, most multiobjective simulated annealing algorithms have the advantage that they allow for a broad search of the solution space at first, before gradually restricting the search so as to decrease the acceptance of non-improving moves later on during the search.

In multiobjective problems, generating neighbouring moves *via* simulated annealing can result in one of three different outcomes: (1) An improving move with respect to all objectives (*i.e.* the current solution is dominated by the neighbouring solution); or (2) simultaneous improvement and deterioration with respect to different objectives (*i.e.* neither solution dominates the other); or (3) a deteriorating move with respect to all objectives (*i.e.* the neighbouring solution is dominated by the current solution). While outcomes (1) and (3) above may also occur in single-objective optimisation problems, outcome (2) only manifests itself in multiobjective optimisation problems.

Because simulated annealing only generates a single solution at every iteration and does not involve a learning component, an *archive* may be employed to record all non-dominated solutions found during the course of the search. All generated moves are candidates for archiving, and are each tested for dominance with respect to every solution in the archive. If a candidate solution dominates any solutions in the archive, these now dominated solutions are permanently removed from the archive and the candidate solution is added instead. If, on the other hand, a candidate solution is dominated by any solutions in the archive, it is not archived, and the archive remains unchanged. Finally, if a candidate solution neither dominates nor is dominated by any solutions in the archive, it is added to the archive without removing any other archived solutions.

The remainder of this section is dedicated to a brief discussion of certain popular multiobjective simulated annealing implementations from the literature. The reader is referred to [138] for a more in-depth discussion on these multiobjective simulated annealing implementations.

The first multiobjective version of the method of simulated annealing was presented by Serafini [127] in 1994, who examined various rules associated with the probability of accepting a neighbouring solution, namely *scalar ordering*, *Pareto ordering* and *cone ordering*. A special rule which aims to concentrate the search exclusively on the non-dominated solutions, and consists of a combination of the above-mentioned rules, was also proposed. Following this pioneering study on multiobjective simulated annealing, many papers proposed unique techniques for tackling various types of multiobjective optimisation problems with the aim of improving algorithmic performance.

In 1998, the so-called *Ulungu multiobjective simulated annealing* algorithm was proposed [138]. The algorithm was designed to accommodate the simultaneous improvement and deterioration of different objectives when transforming candidate solutions. The probability of accepting a move was calculated by taking the distance between the two solutions into account using a *criteria scalarising* approach. According to this approach, a solution in multidimensional objective space is projected to a unidimensional space using a pre-defined, diversified set of uniformly generated random weight vectors. During the process, a list of solutions is constructed which are not dominated by any other solutions evaluated throughout the search (*i.e.* the archive).

Czyżżak and Jaszkiewicz [35] undertook to modify the Ulungu multiobjective simulated annealing algorithm by proposing a method for combining unicriterion simulated annealing with a genetic algorithm, together called the *Pareto simulated annealing* algorithm. According to this approach, the concept of neighbourhood acceptance of new solutions was based on a cooling schedule (obtained from the method of simulated annealing) combined with a population sample of solutions (obtained from the genetic algorithm). At each iteration, the objective weights used in the acceptance probability of neighbouring solutions were tuned in a certain manner to ensure that the solutions generated cover the entire non-dominated front. In other words, the higher the weight associated with a certain objective, the lower the probability of accepting a move that decreases the value of this objective and, therefore, the greater the probability of improving that objective from one generation to another. The use of a population exploring the search space configured by the annealing procedure ensured that a large and diversified set of good solutions was uncovered by the time the system reached a solidified state.

In 2000, Suppapitnarm *et al.* [139] combined the notion of *archiving* with a new systematic technique for periodically restarting the search from a carefully selected archived solution according to a subprocess called the *return-to-base* period strategy. The return-to-base period refers to a pre-determined, fixed number of iterations during which the search process is calibrated independently of the cooling schedule and independently of the acceptance probability function. The aim of this technique is to expose trade-offs between objectives as much as possible. Solutions that are isolated from other archived solutions are favoured as return-to-base candidates. In addition, extremal solutions are also favoured as candidates, as these reside most probably at the limits of feasibility, making the search space around them difficult to access otherwise. The return-to-base strategy is only activated once the basic elements of the trade-offs between objectives have developed sufficiently (a good rule of thumb is as soon as the temperatures are first lowered). Thereafter, the return-to-base rate is increased in order to intensify the search around the archived solutions. A new acceptance probability formulation based on an annealing schedule with multiple temperatures (one for each objective) was also proposed in this approach: if a generated solution is dominated by an archived solution, it is accepted based on the product of changes in all objectives combined.

According to Smith *et al.* [130], however, adapting simulated annealing algorithms to multiobjective optimisation problems by combining the objectives into a single objective function either damages the rate of convergence, or causes the search to potentially restrict its ability severely with respect to fully exploring the non-dominated front. Consequently, Smith *et al.* [130, 131] proposed a dominance-based multiobjective simulated annealing approach which utilises the relative dominance of a solution as part of the acceptance probability function, thereby eliminating the problems associated with composite objective functions. This approach promotes the search towards and across the Pareto front, maintaining the convergence properties of a single-objective annealer, while encouraging exploration of the full trade-off surface, by incorporating the archiving and return-to-base strategies of [139].

2.4 Evolutionary algorithms

> *The original mechanisms of evolution of the living beings rest on the competition which selects the most well adapted individuals to their environment while ensuring a descent, as in the transmission of the useful characteristics to the children which allowed the survival of the parents.* — C. Darwin [19]

Evolutionary computations are stochastic search techniques that imitate the genetic evolutionary process of species based on the Darwinian concepts of *natural selection* and *survival of the fittest*. That is, in nature, evolution occurs as a result of the competition among individuals for scarce resources, resulting in the fittest individuals dominating the weaker ones. Those individuals best adapted for survival are selected for reproduction in order to ensure that strong, useful genetic material is transmitted to their offspring, thus promoting the continuation of the species [24, 48]. A *genetic algorithm* is a population-based metaheuristic in the class of evolutionary algorithms which operates analogously to the concept of evolution *via* natural selection. This algorithm was first introduced by Holland [68] in 1975 in an attempt to understand the (then not-so-popular) mechanisms of self-adaptive systems.

2.4.1 Algorithm outline

In a genetic algorithm, a *population* of *individuals* representing candidate solutions to an optimisation problem is iteratively evolved toward better solutions. A *fitness function* measures the performance of a candidate solution or individual, and is a quantisation of its desirability of being selected for propagation. Each individual is genetically represented by a set of *chromosomes* (or a *genome*) made up of *genes* which may take on a certain range of values from some genetic domain, known as *alleles*. The position of a gene within its chromosome is identified by means of a *locus*. The population evolves in an iterative manner over a certain number of *generations* until a stopping criterion is met. Individuals are represented as strings corresponding to a biological *genotype* (encoded), which defines an individual organism when it is expressed as a *phenotype* (decoded).

Inspired by nature, genetic algorithms attempt to improve the population of individuals at every generation through repetitive application of a range of genetic operators. During every generation, a proportion of *parent* solutions from the current population is chosen to produce a new generation of *offspring* solutions. Parent solutions are selected for *reproduction* through a fitness-based process called *selection* in such a manner that fitter solutions have a higher probability of being selected to reproduce. Offspring solutions are generated *via* a combination of two operators, namely a *crossover* operator, which exchanges genetic material in the form of chromosome sub-strings of a pair of parent solutions to produce one or more offspring solutions (analogous to the process of species reproduction in nature), and a *mutation* operator, which alters a gene in the chromosome string of an offspring solution with a certain probability so as to form a mutated individual. In addition, a *replacement* operator filters the individuals to move on to the next generation so as to maintain a certain population size, after which the entire process is repeated. Record is kept of the fittest individual found throughout the search, which is again called the incumbent solution (similarly to the incumbent solution recorded in the method of simulated annealing).

Populating the next generation mostly by means of new offspring solutions can, however, cause the loss of the best individual(s) from the previous generation. To remedy to this disadvantage, an *elitism* operator may be introduced, which ensures that at least one best solution from the

current parent population is retained (without changes) to the next generation, so that the best solution(s) found throughout the algorithm can survive until the end. Examples of other genetic operators include *immigration*, where one or more randomly generated solutions are introduced into the population during every generation; *education*, where a local transformation is intentionally conducted in respect of certain offspring solutions to try improve their fitness prior to initiation of the selection process or to cure possible infeasibilities; and *inbreeding*, where two highly fit offspring solutions generated from the same pair of parent solutions intentionally reproduce with one another during the following generation.

Although genetic algorithms typically use a subset or variation of genetic operators, the general concept nevertheless remains the same in all cases: genetic algorithms are motivated by the basic idea that populations improve in a way that the average fitness of the population of solutions generally increases during every generation, which consequently increases the chances of finding a global optimum to the optimisation problem at hand after a large number of generations. A flowchart of the general working of a genetic algorithm is presented in Figure 2.4, where only the fundamental genetic operators are considered, and a pseudo-code description of the basic steps featured in this kind of search is given in Algorithm 2.2.

Algorithm 2.2: Genetic algorithm outline

```
1  Generate initial population of individuals
2  Evaluate the fitness of each individual
3  while Stopping criterion not met do
4      Select parent solutions for reproduction
5      Generate offspring population
6      Apply mutation operator to offspring solutions
7      Evaluate fitness of offspring solutions
8      Select individuals to be carried over to the next generation
9      Update incumbent solution (if necessary)
10 Report incumbent solution
```

2.4.2 Multiobjective evolutionary algorithms

Like numerous other search techniques, multiobjective evolutionary algorithms were initially designed as a means to find trade-offs between the objective performance measures of candidate solutions. The overall benefits that evolutionary algorithms bring to multiobjective decision making problems are today the reason for the growing interest of researchers in this field, in particular with regard to their ability to generate several elements of the Pareto optimal set in a single run [81]. It is believed that evolutionary algorithms and, in particular, genetic algorithms, are so well suited for these types of problems as a result of their analogous connection to biological processes exhibiting multiobjective features in nature. According to Zitzler *et al.* [167], two major problems must be addressed when solving a multiobjective optimisation problem by means of a genetic algorithm. One is to configure the fitness assignment and selection processes effectively in order to guide the search towards the Pareto front, and the other is to maintain a diverse population in order to prevent premature convergence and generate a good spread of solutions along the approximate Pareto front.

During the period 1993–1995, researchers began in earnest to investigate the possibility of using evolutionary algorithmic techniques for solving multiobjective optimisation problems. Among these early approaches, the *multiobjective genetic algorithm* [105], the *non-dominated sorting*

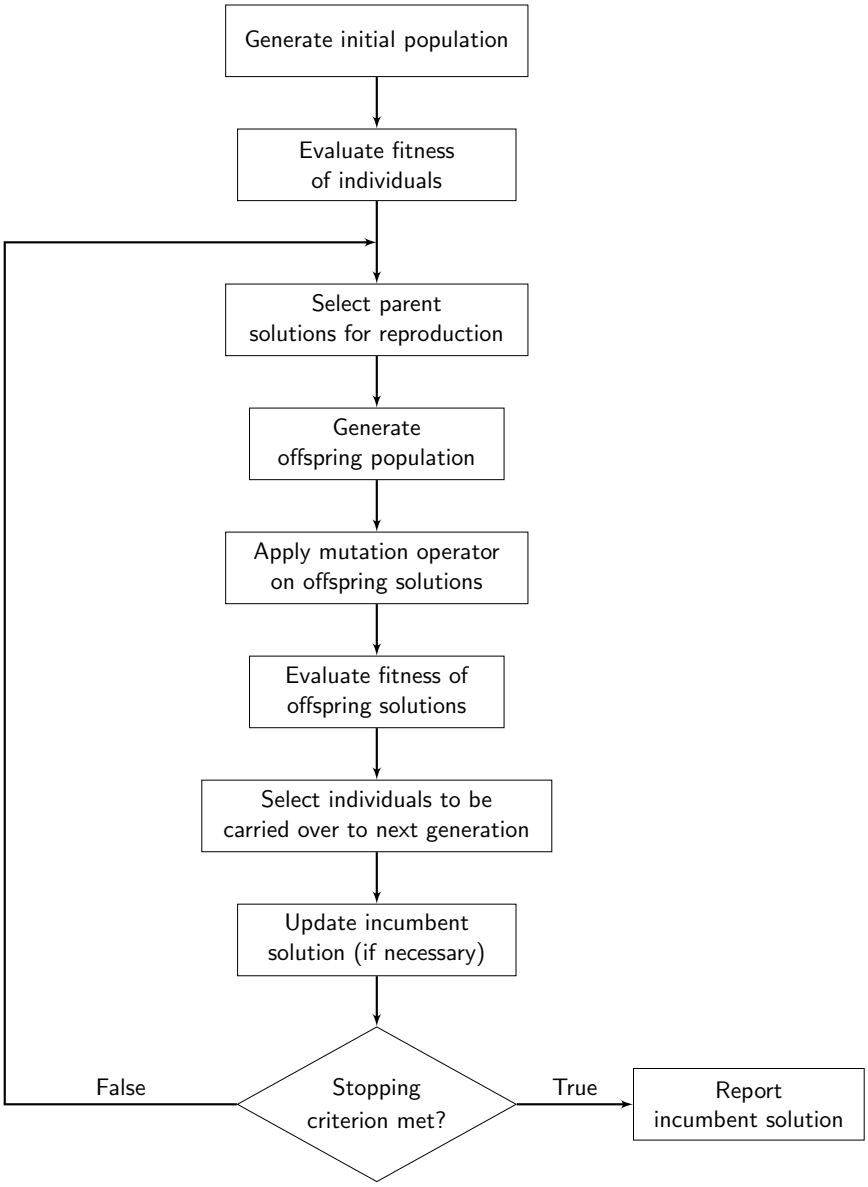


FIGURE 2.4: *Flow of events in a genetic algorithm.*

genetic algorithm (NSGA-I) [134] and the *niched Pareto genetic algorithm* [69] received particular attention. These algorithms were designed with the necessary additional components to convert single objective evolutionary algorithms to multiobjective counterparts without the use of weighting techniques, which often induce inconsistencies. Moreover, they are able to solve problems in multidimentional objective space and generate multiple non-dominated solutions near/along the Pareto front in a single run for a wide variety of optimisation problems. In general, two fundamental features are found in such techniques: (1) the assignment of fitness values to population individuals based on *non-dominated sorting* and (2) the preservation of diversity among solutions located on the same non-dominated front.

Researchers have, however, realised that, although the above-mentioned approaches are perhaps surprisingly effective in finding non-dominated solutions for many test problems, a wider variety of operators than mere crossover and mutation are required to improve computational performance. For example, elitism was introduced in an attempt to improve the convergence properties of the algorithms and to prevent the extinction of good solutions once they have been uncovered. The suggestion to include elitism was indeed a good one, as it has been shown that multiobjective evolutionary algorithms are subsequently able to achieve better converging properties [167], and are generally able to find non-dominated solutions faster. The *strength Pareto evolutionary algorithm* [169] and the *Pareto archived evolution strategy* [79], in particular, are popular multiobjective genetic algorithm techniques which make use of the notion of elitism.

The NSGA-I mentioned above was one of the very first multiobjective evolutionary algorithms to function without the use of user-defined objective preference weights (which amounts to solving a multiobjective problem in uni-dimensional objective space). In other words, the non-dominated population of the last generation represents an approximation of the Pareto-optimal set, from which a compromise solution may be chosen at the discretion of the decision maker. During the 1990s, the performance level of this algorithm was very high relative to the alternative, more traditional methods then available. Continual research in the field of multiobjective evolutionary algorithms over time, however, allowed for the discovery of shortcomings in the method. More specifically, three important criticisms were raised. The first had to do with the high computational complexity of the NSGA-I. Implementation of the algorithm is computationally expensive for large population sizes, particularly due to the complexity associated with its non-dominated sorting procedure, which has to be performed at every generation. Secondly, the NSGA-I lacked the incorporation of elitism. As mentioned earlier, not only does elitism improve the convergence rate of the algorithm, it also prevents the loss of good solutions once they have been uncovered. Lastly, it was argued that the NSGA-I requires an additional parameter for a more effective diversity-preservation mechanism. In particular, the NSGA-I incorporated a so-called *sharing parameter* to this effect, which may be thought of as the distance metric chosen to calculate a proximity measure between two individuals in the current population. However, the addition of such a parameter raises two main complications. First, the performance of the sharing function in terms of maintaining a good spread of solutions relies too heavily on the parameter value chosen (which is user-defined). Secondly, according to this diversity preservation mechanism, every solution in the current population must be compared with every other solution, which may dramatically increase the problem complexity when the size of the population is large (in addition to the complexity associated with the non-dominated sorting procedure pointed out above).

Seven years later, Deb *et al.* [41] developed an improved version of the original NSGA-I, and called it the *NSGA-II*. Their method incorporates a better sorting algorithm, allows for elitism, and does not require the user to define a sharing parameter *a priori*, hence addressing all of the disadvantages raised in respect of the original NSGA-I. Perhaps the main difference between

the two algorithm versions is that, while the fitness of a candidate solution in the NSGA-I is evaluated using non-dominated sorting only, the NSGA-II makes use of non-dominated sorting *as well as* diversity preservation in order to assess the fitness of individuals. Moreover, it has been demonstrated that the NSGA-II outperforms two other successful multiobjective evolutionary algorithms (namely the Strength Pareto Evolutionary Algorithm and the Pareto Archived Evolution Strategy) on the grounds of not only finding a more diverse set of candidate solutions but also in achieving convergence in close proximity to the true Pareto-optimal set for a wide variety of problems.

Overall, multiobjective evolutionary algorithms generally share certain fundamental common features. Their differences are not as complex as one might expect; they are often just individually designed incarnations of the same basic principle achieved in a unique, creative manner. According to Coello *et al.* [29], the generic goals and operator design of all these algorithmic variations are (1) to preserve non-dominated points in objective space and respective solution points in decision space, (2) to continue to make algorithmic progress toward the Pareto front in objective space, (3) to maintain diversity among the points on or near the Pareto front, and (4) to provide the decision maker with a sufficient, but not overwhelming number of non-dominated points for alternative selection.

2.4.3 Fitness assignment and diversity preservation

As previously discussed, one goal in multiobjective evolutionary optimisation is to ensure that higher quality individuals have a greater chance of being selected to pass their genetic material on to the next generation. Moreover, it was mentioned in §2.4.1 that the quality of a solution in a genetic algorithm is assessed by its fitness level. In single-objective optimisation problems, the fitness values of individuals are typically assessed on a uni-dimensional continuum, where individuals with higher fitness values are deemed more desirable. In multiobjective optimisation problems, however, the notion of dominance (see §2.1.4) is typically used to assess the fitness of individuals (such fitness measures are thus represented as cardinal numbers), in contrast to single-objective optimisation problems, where the fitness of a solution is purely based on its objective function value along the real number line. Perhaps the two most popular dominance criteria used in the design of a fitness measure in multiobjective genetic algorithms are *dominance strength*, which represents the number of solutions that a particular individual dominates in a population, and *dominance count*, which is the number of solutions in the population that dominate a particular individual solution [120].

Another goal in multiobjective evolutionary optimisation is to maintain diversity among population members. This may be achieved by considering the *density* of individual solutions in objective space, in the sense that solutions in less crowded regions are favoured to be selected for reproduction and/or carried over to the next generation in an attempt to stimulate exploration of unexplored regions of the search space. *Restricted mating*, *crowding distance* and *relaxed domination* are examples of popular diversity preservation measures [120].

2.4.4 Genetic operators

A brief overview of some fundamental genetic algorithm operators, namely selection, crossover and mutation, is presented in this section. Because these operators are applicable to both single and multiobjective problems, no differentiation is made between these two classes of optimisation problems in this section.

Selection operators

Selection refers to the process of selecting parent solutions from a population based primarily on their fitness values and, secondarily on their relative density values, to form a *mating pool* from which offspring solutions are created. In evolutionary algorithms, *selective pressure* is a term widely used to characterise the high emphasis on selection of the best individuals [3]. Selection pressure may be interpreted as the driving force behind improving the average population fitness over the course of generations which, in turn, has an impact on the convergence properties of the algorithm. Because selection is completely independent of the remaining operators applied in the algorithm, it may be characterised as a universal, problem-independent, operator. According to Back [3], the impact of the control parameters on the selection pressure should be simple and predictable to some extent. Additionally, a single parameter is preferred for the implementation of selective pressure, and the range of selective pressure that may be achieved by varying this control parameter should be as large as possible.

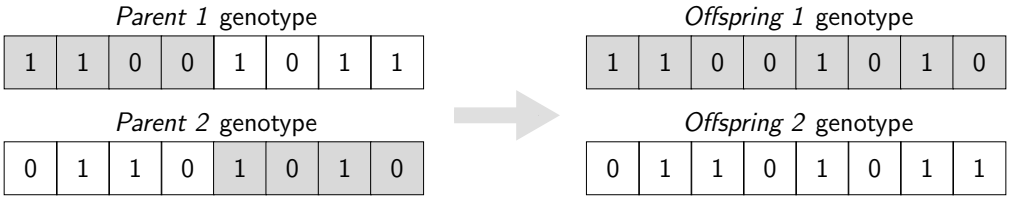
Many types of selection operators are described in the literature, all essentially with the single collective purpose of selecting individuals from the population in a probabilistic manner and inserting them in the mating pool. Examples of selection operators which may be employed in evolutionary algorithms include *fitness proportion selection* (also known as *roulette wheel selection*), where the probability of an individual being selected is simply assessed as the quotient of its fitness level and the sum of the fitness levels of all individuals in the population; *stochastic universal sampling*, where weaker individuals are given a fairer chance of being selected than in fitness proportion selection; *tournament selection*, where a certain number of individuals chosen at random from the population compete with one another in terms of fitness levels from whence a victor is selected[8]; and *truncation selection*, where the population members are ranked by fitness, and a certain proportion of the fittest individuals are all selected, each to reproduce a certain number of times. The interested reader is referred to [81] for a description on the workings of these selection procedures.

Crossover operators

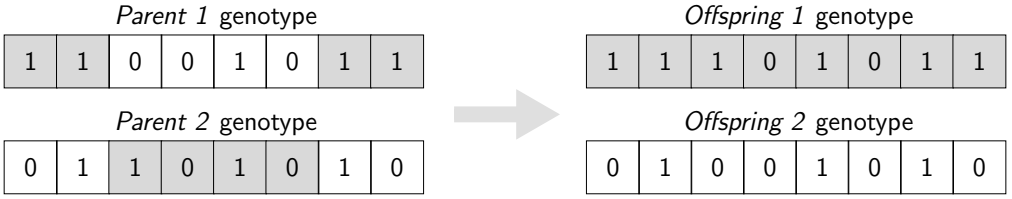
The crossover genetic operator combines the chromosomes of two parent solutions in order to produce one or more offspring solutions. The motivation behind the use of this operator is that an offspring solution might turn out better than both of its parents if it acquires good sets of genes from each of them. There exist many types of crossover operators in the literature, and the type of operator to be implemented depends mostly on the type of optimisation problem at hand and, in particular, on the nature of the solution representation scheme adopted.

Examples of crossover operators which may be employed include *single point* crossover, in which all genetic data beyond the crossover point in each chromosome string is swapped between the two strings; *two-point* crossover, in which all genetic material between two crossover points in each chromosome string is swapped between the two strings; *cut-and-splice* crossover, in which each parent solution has a separate crossover point, causing the resulting offspring chromosome strings to vary in size; and *uniform* crossover, which may be viewed as a multipoint crossover in which the number of crossover points is unspecified *a priori*. The interested reader is referred to [81] for a more thorough description on the workings of these crossover procedures. Graphical illustrations of the first three crossover operators listed above may be found in Figure 2.5 for the case where the chromosome strings are represented as binary coded genotypes.

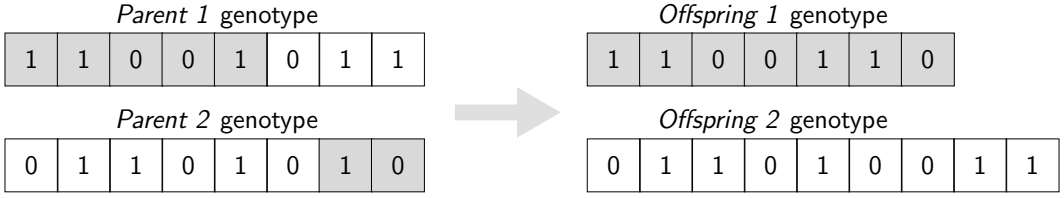
[8] With tournament selection, selection pressure is adjusted by altering the tournament size, where a larger tournament size affords weaker individuals a lower chance of being selected.



(a) *Single point crossover.*



(b) *Two-point crossover.*



(c) *Cut-and-splice crossover.*

FIGURE 2.5: *Illustration of three popular crossover procedures in the literature.*

Mutation operators

> *The individuals having the best fitnesses are reproduced more often than the others and replace the worst ones. If the variation operators are inhibited, the best individual should reproduce more quickly than the others, until its copies completely take over the population.* — J. Dréo [48]

Mutation is often considered a minor operator in genetic algorithms. It is aimed at maintaining a minimum level of diversity among population individuals, which the crossover operator may perhaps not be able to ensure at all times, in an attempt to move the search toward unexplored regions of the search space with the hope of finding high-quality solutions there. A mutation operator typically modifies an offspring solution stochastically in such a way that the result of the transformation is close to its original chromosome form (*i.e.* a local search is performed). The proportion of mutated individuals in the offspring population is controlled using a *mutation rate* parameter. A mutation operator with a significantly high mutation rate is expected to produce the desired results in terms of preservation of diversity, particularly in respect of preventing population individuals from becoming too similar to one another. As a result, it is expected to slow down the evolution progress to a certain extent. Moreover, according to Dréo [48], mutations may be particularly useful for significantly improving the quality of solutions discovered when most of the population is located in the neighbourhood of a global optimum or Pareto front (whereas the importance of the crossover operator diminishes once the population reaches that stage in the algorithm), provided that the mutation rate is high enough. Setting the mutation

rate too high, however, may result in individuals being produced almost independently from the genetic material passed on to them from their parents, which may cause the algorithmic progress to mimic a collection of primitive random walks in the solution space.

Mutation is typically performed by altering the state of a gene in a chromosome string according to a certain probability, which is obtained from the mutation rate parameter. As with crossover operators, the type of mutation operator to be implemented depends mostly on the type of optimisation problem at hand. Examples of mutation operators which may be employed include *bit string* mutation, where alleles of binary coded genotypes are each altered according to a certain probability based on the length of the chromosome string and the mutation rate; *boundary* mutation, where alleles are replaced by either one of user-defined upper or lower bounds according to a certain probability; *uniform* mutation, where alleles are replaced by a uniform random value selected between user-defined bounds according to a certain probability; and *non-uniform* mutation, where the mutation rate is altered throughout the algorithm in order to prevent the population from stagnating during the early stages of evolution while increasing the expected number of mutated offspring solutions during later stages of the process. An example of bit string mutation is shown in Figure 2.6. The interested reader is referred to [81] for a more detailed description of the workings of the mutation operators mentioned above.

FIGURE 2.6: *Illustration of a bit string mutation operator.*

2.5 Other popular metaheuristics

In this section, three other popular metaheuristics are briefly discussed for single objective optimisation problems, namely the methods of tabu search, ant colony optimisation and particle swarm optimisation. Unlike the previous two sections, in which detailed algorithmic descriptions were provided, the purpose of this section is rather to describe the conceptual ideas behind these innovative search techniques in order to provide more insight into three modern, high-performance metaheuristics.

2.5.1 The method of tabu search

Tabu search is primarily a deterministic local search method which is based on the idea that a short-term memory component should be used during the exploration of the neighbourhood around a candidate solution in an optimisation problem. The goal of incorporating such a memory component in the search is to prohibit it from periodically returning to recently visited solutions and, in particular, to avoid the search from revisiting local optima [48].

The local search process conducted in a tabu search evolves around the *neighbourhood set* $\mathcal{N}(\boldsymbol{x})$ of a solution $\boldsymbol{x}$. This set contains all feasible solutions obtainable from $\boldsymbol{x}$ by applying a move from a so-called *move set*. This move set traditionally consists of a finite number of solution transformation methods. In general, however, a tabu search does not need to evaluate the full neighbourhood set of a solution when this set is large[9], but is rather expected to make an

[9] In contrast, the method of simulated annealing only considers one neighbouring solution at every iteration.

intelligent choice with respect to the nature of the generated neighbourhood set. Hence, the purpose of a move set is to reduce the size of the neighbourhood around a solution in order to prevent the search from visiting neighbouring solutions not worth visiting. Alternatively, restrictions may be imposed on move instances, rather than specific solutions, whose reversals are thereby forbidden. A strong tabu search is therefore required to include, amongst other aspects, an effective local search mechanism. In particular, the relationship between the quality of the move set, the types of moves and the computational budget necessary for their evaluation in objective space, ought to be configured in the correct manner.

Following the creation of a neighbourhood set around a current solution, the best solution within that set is then selected as the new current solution. To prevent the search from revisiting a recent solution, a *tabu list* $\mathcal{T}$ is maintained which keeps track of the $|\mathcal{T}|$ previously visited solutions, called *tabu solutions*. These solutions are excluded from the neighbourhood set of the current solution as long as they remain members of the tabu list. This list is usually managed to maintain a fixed length in a *first-in-first-out* fashion, after which tabu solutions removed from the list are relabelled as non-tabu. There exist several stopping criteria that may be employed to terminate the algorithm.

A flowchart of the working of the method of tabu search is presented in Figure 2.7. In addition, a pseudo-code description of the basic steps featured in a tabu search is given in Algorithm 2.3. In line 8 of the pseudo-code, the previous solution is added to the head of the tabu list, while the solution at the tail of the list is reclassified as being non-tabu, provided that the list has already reached its full capacity (if that is not the case, then the tail in this list may be thought of as an empty slot).

Algorithm 2.3: Tabu search algorithm outline

```
1  Generate initial feasible solution
2  Set tabu list 𝒯 ← ∅
3  while Stopping criterion not met do
4      Generate current solution neighbourhood 𝒩(𝒙)
5      Evaluate the performance of non-tabu solutions in 𝒩(𝒙)
6      Select best neighbour solution 𝒙*
7      Update incumbent solution (if necessary)
8      Update tabu list 𝒯 ← {𝒯\𝒯(|𝒯|)} ∪ 𝒙
9      Set new current solution 𝒙 ← 𝒙*
10 Report incumbent solution
```

2.5.2 Ant colony optimisation

The method of ant colony optimisation is a so-called *self-organisation* metaheuristic naturally inspired by the collective behaviour of ants whilst seeking food sources. In nature, a colony of ants communicate indirectly with one another *via* ground trails of odorous volatile substances, known as *pheromones* [48]. The ants collectively use these trails to indicate desired channels between their nest and a food source for exploitation, enabling the colony to ultimately assess an optimal path towards the source without requiring individual ants to have a global pattern overview (*i.e.* only local information is utilised by the ants). These indirect social interactions by means of modifications of the environment is better known as *stigmergy* in biological studies.

A trail containing a higher concentration of pheromones is deemed more desirable and has

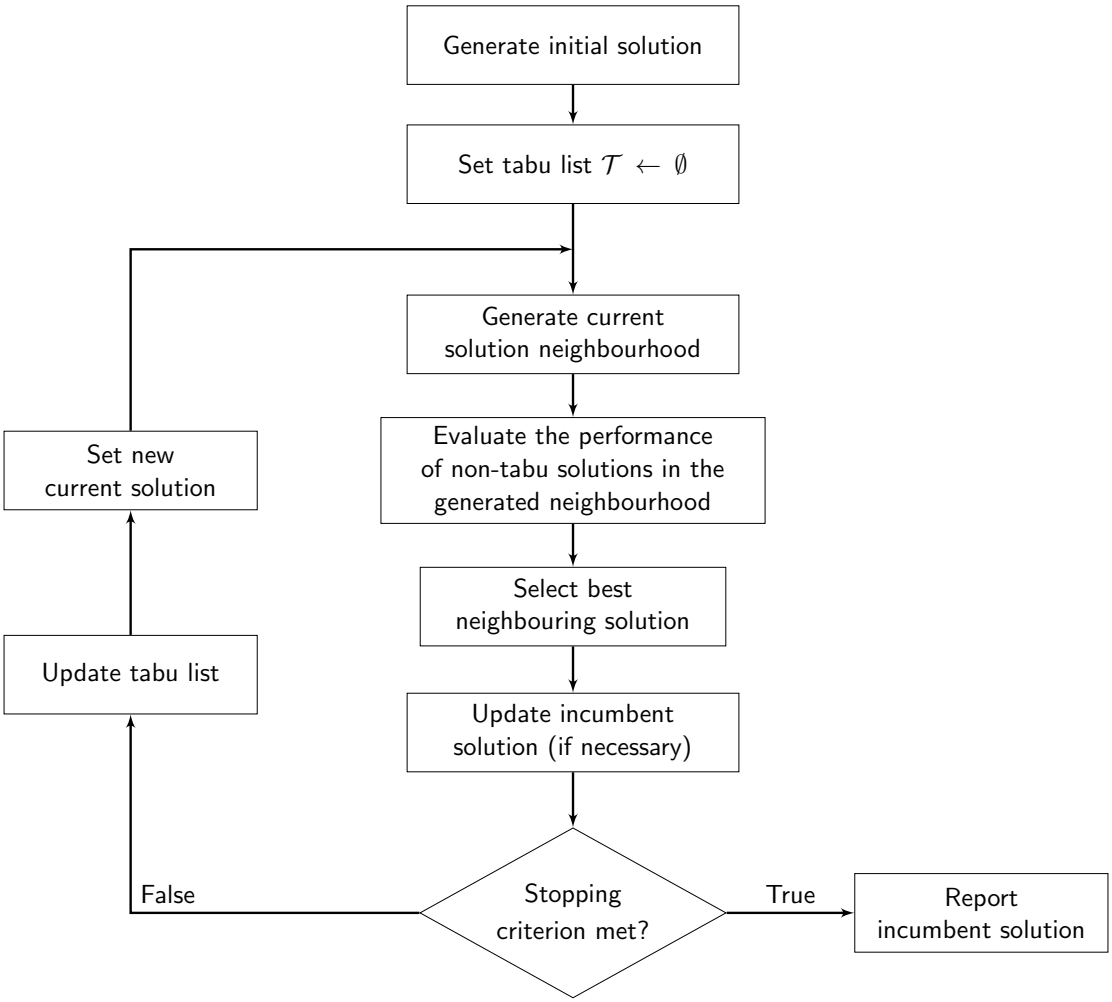


FIGURE 2.7: *Flow of operations in the method of tabu search.*

a higher probability of being traversed by any given ant. In addition, the natural chemical constitution of pheromone levels causes these trails to evaporate over time, unless other ants re-traverse the trails and deposit more pheromones onto them. This is a necessary phenomenon in the process as it allows the system to "forget poor solutions" and avoid being trapped at local optima. The concentration of pheromones along a certain trail is therefore an indication of the volume of ant traffic on that trail. As a result, the path with the highest concentration of pheromones (typically the shortest path) will be chosen by the majority of ants in the long run until the food source expires. This method has proven to be very popular for solving variants of the *travelling salesman problem* (which initially motivated the design of the method of ant colony optimisation by Dorigo and Gambardella [47] in 1997), as well as routing and graph colouring problems later on.

An ant colony system may be modelled as a complete graph $G(\mathcal{V}, \mathcal{E})$ with a set $\mathcal{V}$ of nodes and a set $\mathcal{E}$ of edges inter-linking all pairs of these nodes, where each ant a in a set of ants traverses the graph and builds a pheromone trail at every iteration $t \in \{1, \ldots, t_{max}\}$ of the process. Each edge is initially assigned a certain pheromone concentration. For each ant at any given node $i \in \mathcal{V}$, the selection of the edge between node i and node j is determined by three factors, namely the list $\mathcal{J}_{ai}$ of remaining nodes to be visited by ant a provided that it is currently at node i,

the reciprocal of the distance between each pair of nodes, called the *visibility* factor, and the concentration of pheromones deposited on each edge in $\mathcal{E}$, called the *trail intensity* factor. An ant moves from its current node toward another node according to a probability distribution constructed according to a so-called *rule of displacement*, which takes these three factors into consideration. In addition, the system uses a list of candidates that stores, for each node, its closest neighbours distance-wise. An ant only considers traversing an edge towards a node in such a list if this node has not previously been explored by the same ant.

The concentration of pheromones on an edge is immediately updated on a local level once an ant has traversed that edge, considering both the trail intensity of that edge as well as the evaporation rate of the pheromone levels. The global update of the system takes place at the end of every iteration and the best feasible solution found is updated in both decision and objective space (*i.e.* the best tour for a travelling salesman problem or the best set of routes for a vehicle routing problem is recorded as the incumbent solution). The edges forming the incumbent solution are then updated on a global level by increasing the pheromone levels of these edges according to a pre-defined rule, thus promoting intensification around the incumbent solution over the next iterations. Because the incumbent solution is updated at every iteration, the termination criterion is generally configured as a pre-specified number of iterations to be performed.

A flowchart of the working of the method of ant colony optimisation is presented in Figure 2.8 and a pseudo-code description of the basic steps featured in this method is given in Algorithm 2.4. It is conceded that the pseudocode in Algorithm 2.4 is not completely generic, but has rather been adapted for travelling salesman problems and routing problems in respect of the number of nodes to be visited in order to form a feasible solution (*i.e.* each node is required to be visited exactly once in such combinatorial optimisation problems).

Algorithm 2.4: Ant colony optimisation algorithm outline

```
1  Assign every edge in G(V, E) an initial pheromone level
2  t ← 0
3  Generate a set A of ants
4  while t < t_max do
5      forall a ∈ A do
6          Assign an initial random node i* ∈ V to ant a
7          J_{ai*} ← V\{i*}
8          ℓ ← 0
9          while ℓ < |V| do
10             Choose a node j ∈ J_{ai} according to the rule of displacement
11             Update the pheromone concentration on edge (i*, j)
12             Update the set of nodes to be visited J_{ai} ← J_{ai*}\{j}
13             i* ← j
14             ℓ ← ℓ + 1
15     Update incumbent solution
16     Perform global pheromone update on incumbent solution edges
17     t ← t + 1
18 Report incumbent solution
```

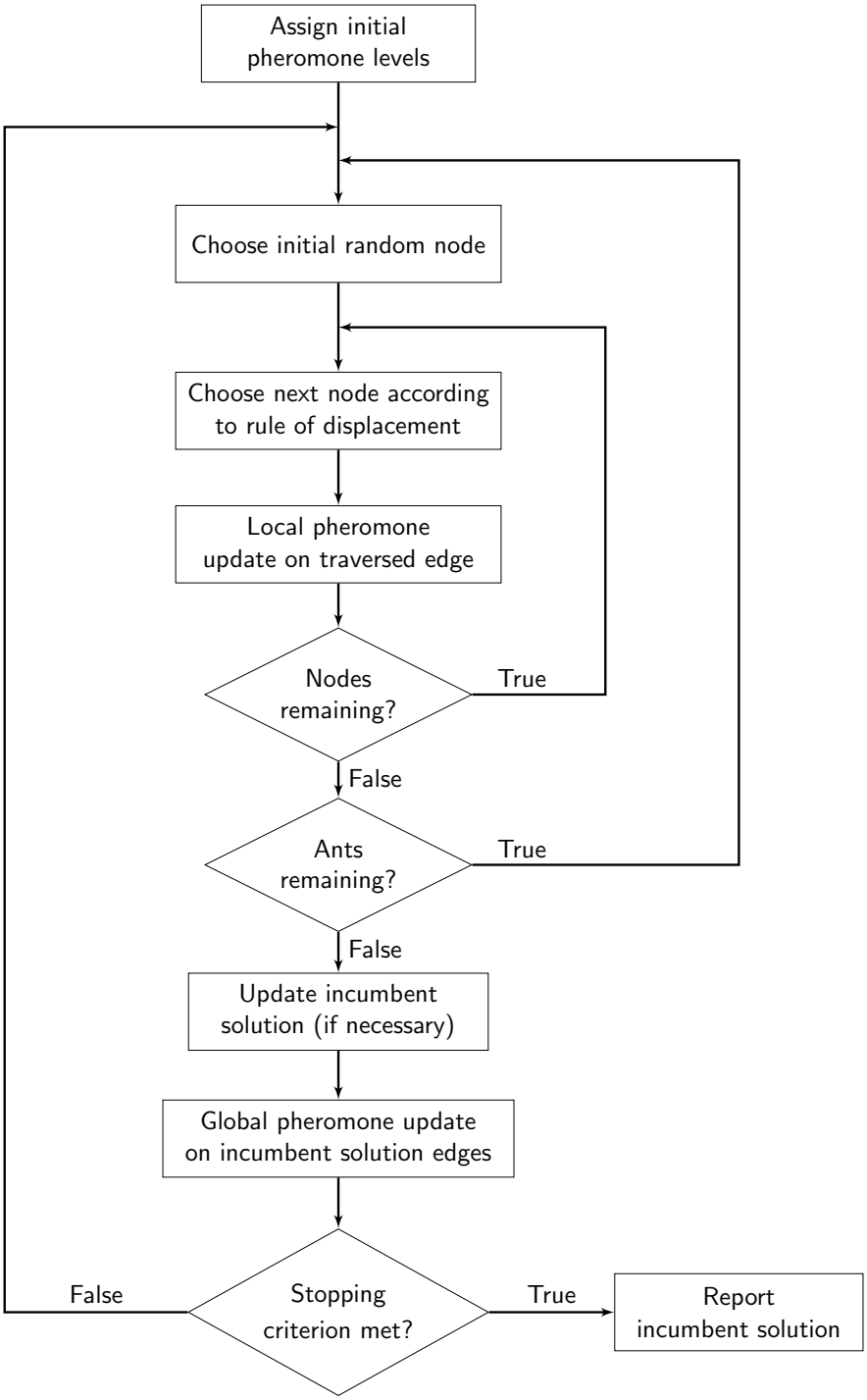


FIGURE 2.8: *Flow of operations in the method of ant colony optimisation.*

2.5.3 Particle swarm optimisation

The method of partical swarm optimisation is a population-based metaheuristic which was originally based on the socio-psychological principles emanating from the collective behaviour of decentralised, self-organised agents [76, 114]. Unlike evolutionary algorithms, particle swarm optimisation does not make use of selection and reproduction operators. Instead, all population members survive throughout the algorithm (as is the case in ant colony optimisation) and their interactions mainly evolve around the iterative improvement of solutions over time. The initial idea behind swarm intelligence was presented by Eberhart and Kennedy [50] in 1995, who attempted to produce computational intelligence by exploiting certain analogues of social interaction, rather than purely being individual-based like most other metaheuristic search techniques.

In the method of particle swarm optimisation, a population of *particles* is initially individually placed at random locations in the solution space of some optimisation problem, and are evaluated in objective space. Each particle in the *swarm* $\boldsymbol{S}$ then follows a path through the solution space. This path is determined by combining its current and historically best locations ("best" with respect to the performance of that location in objective space) with those of one or more other particles in the swarm, as well as allowing for random perturbations.

At every iteration, each particle $\boldsymbol{p}$ is moved one step further from its current location $\boldsymbol{p}_\ell$ according to a certain velocity vector $\boldsymbol{p}_v$. The coordinates of $\boldsymbol{p}_v$ are added to the coordinates of $\boldsymbol{p}_\ell$ to determine the new location of the particle. The best location $\boldsymbol{p}_{\ell^*}$ found thus far during the search for a specific particle is updated at every iteration. In addition, the best location $\mathcal{N}(\boldsymbol{p})_{\ell^*}$ found thus during the search in the neighbourhood of a specific particle is tracked by the particle swarm optimiser. Finally, an incumbent solution representing the best location found thus far by the entire swarm is also archived (*i.e.* the best location found when any particle takes all swarm particles as its topological neighbours). The particle swarm optimisation concept consists of tuning the velocity vector of each particle toward $\mathcal{N}(\boldsymbol{p})_{\ell^*}$ and toward the incumbent solution at every iteration. This sudden change in velocity is magnified by random terms generated for accelerating particles toward these locations [165].

The swarm is more than just a collection of particle solutions. A particle by itself has almost no power to influence the search direction; improvement in the search only occurs when the particles interact with one another. The idea is that, similarly to a flock of birds seeking food sources, the swarm will eventually approach a global optimum of the optimisation problem at hand. A flowchart of the working of the method of particle swarm optimisation is presented in Figure 2.9 and a pseudo-code description of the basic steps featured in this method is given in Algorithm 2.5.

2.6 The vehicle routing problem

The VRP is considered one of the most important and complex combinatorial optimisation problems in the operations research literature. It is easy to describe, but very difficult to solve [135], and is itself a combination of two other celebrated combinatorial optimisation problems, namely the *bin packing problem* in which, given the capacity of a bin and a finite set of items of specific sizes, the minimum number of bins required to contain all the items is sought, and the *travelling salesman problem* in which, given a weighted complete graph, a shortest closed tour including every node in the graph is sought. Perhaps the most appealing feature of the VRP is that its original formulation, known as the *capacitated* VRP, may be extended to nu-

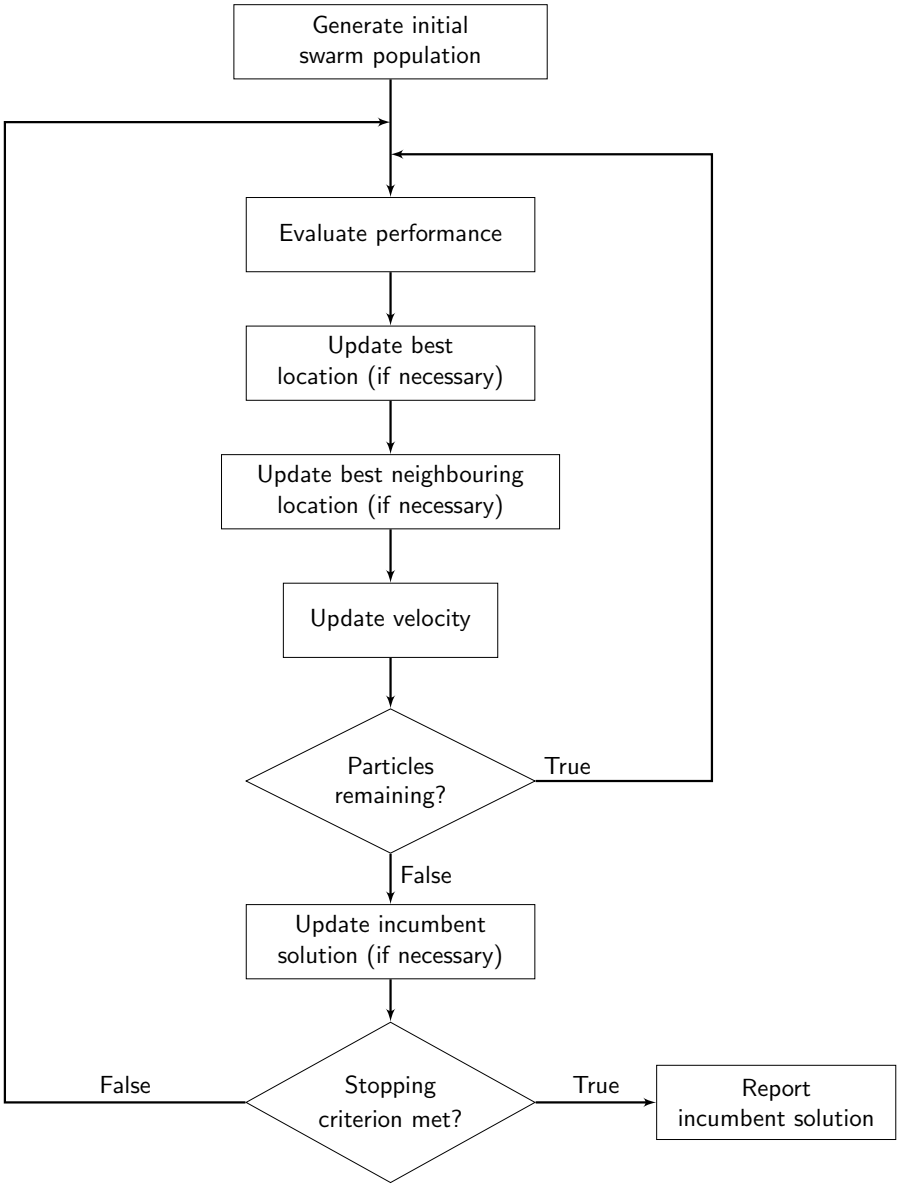


FIGURE 2.9: *Flow of operations in the method of particle swarm optimisation.*

Algorithm 2.5: Particle swarm optimisation algorithm outline

```
1  Generate initial feasible swarm population with random locations and velocity vectors
2  while Stopping criterion not met do
3      forall p ∈ S do
4          Evaluate the performance of the particle in objective space
5          Update best location of the particle (if necessary)
6          Update best location in the neighbourhood of the particle (if necessary)
7          Update velocity vector of the particle
8          Accelerate the particle toward a new location
9      Update incumbent solution (if necessary)
10 Report incumbent solution
```

merous variations for solving a variety of real-world routing problems. This section contains a brief review on the origin and variation of VRPs, and this is followed by a discussion on the nature of specific problem information associated with VRPs. In this discussion, dynamic and stochastic VRPs are afforded special attention. Additionally, a specialised review of solution search methodologies for VRPs related to the MLE response selection problem is conducted.

2.6.1 The origin and variations of the problem

The first description of the VRP was given by Dantzig and Ramser [37] in 1959 in a paper called the *Truck dispatching problem*. The reason for investigating this type of problem was to propose a generalisation of the travelling salesman problem in which the salesman is required to return to his starting point after visiting a certain number of customers. In generic terms, the problem was to find a shortest set of tours which together pass through all vertices of a weighted complete graph such that all tours contain a pre-defined, single point (called the *depot*) in common and the total tour length is a minimum. A real-life problem involving the optimal routing of a fleet of fuel delivery trucks between a fuel depot and a large number of petrol service stations supplied by the depot was then formulated in the 1959 paper of Dantzig and Ramser [37], in which it was assumed that information related to the shortest path between any two nodes in the set of petrol service stations and the fuel demand of each station are known *a priori*. In this scenario, each petrol station was required to be visited exactly once by a single delivery truck, and no truck's cargo was allowed to exceed its pre-determined capacity. Moreover, the routing was to occur in such a way that the fuel demand quantities are all met and such that the total distance covered by the fleet of trucks is a minimum. A procedure based on a linear programming formulation for obtaining a near-optimal solution was proposed as an approximate approach toward solving this problem.

This method, however, has a significant shortcoming with respect to satisfying the minimisation of total routing costs. Indeed, according to this procedure, the distribution routes are prioritised based on the combined demands of a set of customers relative to a certain fraction expressed in terms of the capacity of the vehicle in use and the total number of customers to be served in the system. In other words, the procedure places more emphasis on filling the trucks as close to their full capacities as possible rather than on minimising the overall distance covered by the fleet[10]. As a result, it is possible for customers located far from each other to be assigned to the same route without any form of penalties.

In 1964, Clarke and Wright [26] published a paper introducing an effective greedy heuristic for solving capacitated VRPs, which later became known as the *Clarke-Wright algorithm*. This method iteratively forms routes by creating links between pairs of vertices, ranked according to their *savings* in distance travelled, while maintaining feasibility. The real-life problem in [37] was then solved using this heuristic in order to demonstrate its reasoning and evaluate its performance. Despite consisting of only 30 customers, the Clarke-Wright algorithm generated a 17% improvement in the objective function value of the problem over the former method by Dantzig and Ramser [37].

Numerous algorithms and greedy heuristics have subsequently been published during the fifty years following the publications of these two seminal papers, as the applications of the VRP to real-life problems grew significantly during that period. Modifications to the capacitated VRP model are routinely introduced in the literature in order to incorporate new, fundamental aspects of some practical problem, such as VRPs with *customer time windows*, *split deliveries*,

[10]In multiobjective VRPs, however, one might be interested in minimising the total distance travelled by the fleet *and* to minimise the size of the fleet in service.

multiple depots, *heterogeneous fleets of vehicles*, *customer profits*, *asymmetric travel arcs*, *demand ranges*, *etc.* The interested reader is referred to the authoritative book by Toth and Vigo [150] for generic formulations of these VRP variants. Many papers also focus on in-depth theoretical analyses of these problems, often putting forward effective search techniques for solving them. Smaller studies have also focused on niched practical aspects of variations on the VRP, such as simulating the impact of service lead time on the costs of deliveries from a centralised fuel depot with a homogeneous fleet of vehicles in a medium-scale, semi-rural environment [34]. In most real-life VRP applications, the problem to be solved is very specific. It is therefore critical to achieve a good understanding of the specifics of the problem at hand, and also of the VRP variants and associated optimisation techniques available in the literature, in order to describe, model and solve the problem in an appropriate and effective manner.

2.6.2 The nature of specific problem information

According to Psaraftis [118], real-world applications of VRP formulations may be classified according to four attributes of the nature of information required to specify problem instances, namely the *evolution*, *quality*, *availability* and *processing* of information. The nature of information in a VRP plays a critical role in the characterisation, modelling and solution search methodology that should be adopted. These attributes, together with their respective values, are shown in Table 2.1.

In the above discussion, the evolution of information refers to the information available to the decision maker during the execution of vehicle routes, which may either be classified as *static* (where the input data are known for the entire duration of the routing process and are not subject to any updates), or *dynamic* (where the input data are not known for the entire duration of the routing process, but are generally revealed or updated as time passes). The quality of information refers to possible uncertainty in respect of the available input data, which may either be classified as *deterministic* (where the input data are known with certainty once made available), *forecasts* (where the input data are subject to revisions as the process evolves), *stochastic* (where certain input data are prescribed as probability distributions or evolve according to certain stochastic processes), or *unknown*. The availability of information may either be classified as *local* (where only certain role players in the routing process are immediately made aware of changes in input data), or *global* (where changes in input data are updated continuously to most or all role players). Finally, the processing of information refers to two main paradigms in respect of the processes followed once new information is made available, which may be classified as either *centralised* (where all information is collected and processed by a central unit), or *decentralised* (where some of the information is processed separately from the central unit).

Evolution	Quality	Availability	Processing
Static	*Deterministic*	*Local*	*Centralised*
Dynamic	*Forecasts*	*Global*	*Decentralised*
	Stochastic		
	Unknown		

TABLE 2.1: *Classification of VRPs with respect to the nature of specific problem information [118].*

2.6.3 Dynamic VRPs

As mentioned in the previous section, a VRP is said to be *dynamic* if input data to the problem are made known to the decision maker in a continual fashion and are updated concurrently with the determination of the set of routes. The decision maker is therefore required to solve the problem in a sequential fashion; that is, to solve part of the problem on the basis of the information already available, and then to resolve parts of the problem as new input data are revealed. What distinguishes most modern VRP variants today (particularly in the context of commodity distribution) from their past counterparts is that information used to determine the nature of the routes is today typically continually revealed to the decision maker as a result of revolutionary advances in information and communication technologies [118]. This advancement in the role that information plays in VRP-related systems, together with the significant increase in computing power at relatively low costs, makes it possible for large amounts of input data to be updated in real time.

In contrast to their static counterparts, dynamic VRPs introduce new model aspects and challenges that increase the complexity of the decision making process as well as the assessment of the overall performance of a given routing strategy. It is, for example, more likely that certain customer requests may be rejected at any given time in a dynamic VRP because of the complication and costs associated with serving them on short notice, particularly with respect to the locations of the vehicles in space at that time. Moreover, the ability to redirect an active vehicle to a new route generally allows for cost savings. Dynamic VRPs are also known to increase the set size of the objective functions in a model. For example, whereas the common objective in static VRPs is to minimise routing costs, dynamic VRPs are more likely to introduce subsidiary objectives such as service levels, throughput achievement, revenue maximisation or minimisation of the overall delay between the placements and servicing of new orders [113].

Dynamic VRPs require making decisions in which a compromise or trade-off is sought between the speed of generating a solution and the overall solution quality; that is, the time invested in searching for good decisions comes at the price of less reactiveness to changes in the input data [118]. This aspect is of particular importance in cases where customers are expected to be serviced shortly after placing an order, and where a good decision must consequently be made as fast as possible. In terms of performance, it is therefore critical to design a model and identify an appropriate search technique that produces satisfactory solutions shortly after having received a significant update in input data.

Some dynamic VRP methodologies also account for simplifying the complexity associated with revisiting the decision making process at deterministic points in time by making use of predefined time frames to bound the time dedicated to each problem instance in the form of a sequence of static VRPs [113]. This approach may significantly reduce stress levels on the decision maker as he is only required to perform a decision making process at known, evenly (or appropriately) spread, points in time. Additionally, it potentially allows more time for finding a better solution to each problem instance (provided that the time frames are large enough). This kind of dynamic VRP methodology is therefore effective for solving VRPs in which customer orders are not subject to a high degree of urgency straight after they have been placed (*i.e.* where a long lead time is allowed), or where there exists no objective function for minimising the combined customer service delay time. The solution strategy proposed in [101], for example, divides a working day into a fixed number of time slices of equal lengths. The input data associated with the processing of new orders (which are generated stochastically in time) which fall within a specific time slice are postponed to the beginning of the following time slice. The decision maker then resolves the problem at the beginning of every time slice by computing a

new set of routes using the input data accumulated during the course of the previous time slice in addition to the (known and still applicable) data gathered prior to that time slice over the course of the working day.

2.6.4 Stochastic VRPs

As mentioned earlier, VRPs in which certain input data are prescribed as probability distributions (*i.e.* whenever one or more of the model parameters take the form of random variables), or evolve with respect to certain stochastic processes, are said to be stochastic. In the literature, VRPs of this type often tend to focus on the management of stochastic input data in such a way as to minimise expected costs of routing operations (in the case of VRPs with stochastic travel distances and/or travel times [83]), minimise expected delivery shortages (in the case of VRPs with stochastic customer demands [64]) and preventing vehicle autonomy route failures[11]. Such planning therefore typically involves designing routes in an economical fashion before actual customer demands become known while minimising the overall probability of route failures.

In stochastic VRPs, unforeseen consequences manifested in real time typically lead to an increase in expenses, both directly and indirectly. In particular, a trade-off exists between the savings in travel costs due to larger routes and the frequency of route failures [136]. These expenses are typically triggered in the form of additional vehicles dispatched to complete unfinished routes, extra-time remuneration of drivers as a result of autonomy failures, customer dissatisfaction due to poor service (where certain customers were never serviced or only partially serviced), waste of time associated with loading and unloading vehicle shipments which were not delivered and waste of fuel caused by vehicles travelling with undelivered (and thus unnecessary) loads.

Two general mathematical modelling approaches are found in the literature of stochastic VRPs (and, in fact, in any stochastic programming problems). The first of these approaches is *chance constrained programming*, where the objective function is to minimise planned route costs while ensuring that having a route duration/distance in excess of the assigned vehicle's time/distance autonomy does not exceed a predetermined threshold. The second approach is *stochastic programming with recourse*, where the cost of repairing a violated constrained to feasibility is directly considered in the problem formulation, and where the objective function consists in minimising the expected routing costs as well as the penalty costs associated with violated constraints. While chance constrained programming is relatively more appealing and easier to use, stochastic programming with recourse is able to take the expected costs related to excess route durations into account as well as in proportion to the extent of excess. In one of the recourse models proposed for the VRP with stochastic travel times by Laporte *et al.* [83], for example, the vehicles incur a routing penalty which is directly proportional to the duration of their respective routes in excess of a (preset) fixed time parameter.

2.6.5 VRP solution methodologies in the literature

Due to their simplicity and fast computations, VRP heuristics are very popular for obtaining satisfactory, though only approximately Pareto-optimal, solutions to small-scale problem instances. Solving large-scale VRP instances with multiple complex objectives is, however, very difficult using only heuristic techniques if the goal is to attain a good approximation of the Pareto front. VRPs of this nature call for the use of alternative, more powerful solution search techniques.

[11] A *route failure* is said to occur whenever the travel time and/or travel distance associated with a route exceeds the assigned vehicle's distance and/or time autonomy levels. Such a route is then said to be *incomplete* or *unfinished*.

There is consensus in the literature that VRPs are computationally very hard to solve, to the point where instances of the capacitated, single-objective VRP (the most basic VRP formulation) with 10 vehicles and more than 30–40 customers can only be solved approximately with the use of metaheuristics [119]. Of course, this computational complexity typically increases when solving more specialised VRP variants, such as solving VRPs with multiple depots in multiobjective space. Moreover, VRPs with a heterogeneous fleet of vehicles are considered much harder to solve than those with a homogeneous fleet [109].

The multi-depot VRP is a variant of the capacitated VRP in which routes are simultaneously sought for several vehicles originating from multiple depots, serving a fixed set of customers and then returning to their original depots. Compared to simpler VRPs and their variants, for which abundant literature exists, only a relatively small volume of research has been done on multi-depot VRPs. A small number of good optimisation techniques have nevertheless appeared in the literature for this problem. Most of these techniques tend to break down an instance of the problem into a series of single-depot sub-instances and/or only solve it for a single objective and/or for a homogeneous fleet of vehicles [66, 67, 109, 140, 157]. Because vehicles always start and complete their routes at the same depots, it is easy to merge the set of customer vertices with that of the depot vertices in such simplified model formulations.

In [66] and [140], for example, the solution process for a single-objective, multi-depot VRP with a homogeneous vehicle fleet is decomposed into a three-phase heuristic approach. The first phase involves grouping the set of customers to be served by each depot (a clustering sub-problem), while the second phase entails assigning customers served from the same depot to several routes so that the capacity constraint associated with the vehicles is not violated (a routing sub-problem). The third phase consists of determining the visitation sequence of customers located on the same route (a scheduling sub-problem). The chief difference between considering multiple depots simultaneously and solving multiple single-depot sub-problems separately is that the local search operators in metaheuristics tailored to these approaches differ significantly in nature [157].

According to Salhi *et al.* [123], however, the decomposition of a multi-depot VRP into several single-depot sub-problems by (naively) assigning the customers to the depots closest to them (distance-wise), and solving these sub-problems individually, using a suitable approach, is easy to carry out, but usually leads to poor, significantly sub-optimal solutions. As a result, the mathematical modelling approach proposed in [123] is perhaps best suited to solving instances of the MLE response selection problem in a centralised or intermediate decision making paradigm (where the problem is solved globally). Salhi *et al.* [123] propose a complete mixed-integer, linear formulation for a generic multi-depot VRP with a heterogeneous vehicle fleet. Moreover, formulation variants are also offered for alternative VRP scenarios. For instance, multi-depot VRP variants are considered in which the number of vehicles of a given type is known (*i.e.* a heterogeneous fleet with defined classes of vehicles); in which certain types of vehicles cannot be accommodated at certain depots; in which a vehicle is not required to return to the same depot from whence it originated; or in which a maximum route length constraint associated with each class of vehicle is imposed (*i.e.* a distance-constrained VRP). Salhi *et al.* [123] then solve the problem (approximately) using a variable neighbourhood search applied to the notion of *borderline customers*, using six different local search operators. Unfortunately, the model formulation and search technique presented in [123] is applicable to single-objective optimisation only. Consequently, it is unlikely that this variable neighbourhood search heuristic may be as effective for solving similar problems in multiobjective space.

Many studies favour the use of genetic algorithms for solving the above-mentioned multiobjective VRP variants, often suggesting specific or unique genetic operators [66, 67, 88, 140, 157]. Moreover, due to the high complexity level associated with problems of this kind, many stud-

ies favour the use of hybrids over more traditional evolutionary computation approaches in an attempt to improve algorithmic performance. The innovative hybrid genetic search with adaptive control proposed by Vidal *et al.* [157], for example, offers an optimisation methodology for solving (among other problems), the multi-depot, periodic VRP with capacitated vehicles and time windows. This hybrid includes a number of advanced features in terms of chromosome fitness evaluation, offspring generation and improvement, and effective population management. Two sub-populations are managed in [157] (one containing feasible individuals and the other infeasible ones), and the mating pool is populated using parents from both sub-populations. Furthermore, the selection operator mechanism takes into account both the fitness of individuals and the level of contribution they provide to the diversity of the population. Offspring solutions are enhanced fitness-wise using so-called *education* and *repair* local search procedures. In [66], another hybrid genetic algorithm is proposed for solving difficult multi-depot VRPs. Here, three external heuristics are combined with a traditional genetic algorithm to design a hybrid between the Clarke-Wright algorithm [26], the nearest neighbour heuristic [145] and the iterated swap procedure [86]. While the former two heuristics are solely used for generating high-quality initial populations, the iterated swap procedure is applied to offspring solutions which, if found to have better fitness values than one (or both) of their parents, replace that (or those) parent solution(s) in the next generation of candidate solutions.

Although not as popular for solving complex multiobjective VRPs, certain authors have also adopted traditional, sequential simulated annealing techniques as a means to finding solutions to less complex VRPs [80, 87, 156, 164]. In these approaches, the very popular *inter-route* and *intra-route* solution transformation techniques [156] are often used to provide the necessary exploration and exploitation features to this metaheuristic for the purpose of generating a good non-dominated front when tuned appropriately. Harnessing the processing power of parallel computing has also been proposed for solving complex multiobjective VRPs [4].

Combining genetic algorithms with simulated annealing is not an uncommon strategy in the literature for both single-objective and multiobjective optimisation VRPs [25, 108, 115, 128]. The hybrid proposed by Chen and Xu [25], for example, first employs the crossover and mutation operations of genetic algorithms to produce offspring solutions, after which the fitness values of the offspring solutions are evaluated, accepting offspring solutions in the next generation of candidate solutions according to the Metropolis acceptance rule of simulated annealing.

Other metaheursitics employed in the literature for solving multiobjective VRPs include tabu search, ant colony optimisation and particle swarm optimisation (see §2.5). Szymon and Dominik [141], for instance, emphasise the value of using a parallel tabu search algorithm for solving a multiobjective distance-constrained VRP. In [5], an ant colony system is put forward for solving a tri-objective VRP in which it is required to minimise the number of vehicles used, the total travel time and the overall delivery time. Finally, in [23], a particle swarm algorithm incorporating certain features from multiobjective optimisation is used to allow particle solutions to conduct a dynamic inter-objective trade-off in order to evolve seemingly poor infeasible routes to very good ones using the benefits of swarm intelligence.

2.7 Dynamic multiobjective optimisation approaches

As discussed in §2.6.3, significant changes in one or more input parameters in a dynamic optimisation problem may cause the currently implemented solution to deteriorate significantly in terms of quality and reliability. A decision maker must therefore solve part of the problem on the basis of the information currently available, and then re-solve part of the problem as

certain new input data are revealed and as fragments of the solution might no longer be feasible or preferred. In this section, various approaches from the literature adopted in the context of solving dynamic multiobjective optimisation problems are discussed in varying amount of detail.

2.7.1 On-line *versus* off-line optimisation

According to Deb *et al.* [40], one of two computational procedures is typically adopted for solving dynamic optimisation problems in general. In one procedure, optimal control laws or rules are transformed over time by solving a so-called *off-line* optimisation problem, configured by evaluating a given solution on a number of potential feasible scenarios to the dynamic problem. The alternative procedure is known as *on-line* optimisation, where the (dynamic) problem is considered stationary over a fixed period of time, and during which a solution search technique is expected to seek an approximation set of the Pareto front. Thereafter, a new problem is constructed based on the current situation, following which a new optimisation process is conducted during the new time period. Dynamic on-line optimisation problems can be classified further into two subproblems in which the problem re-initialisation and optimisation procedures may either occur at known, regular intervals in time, or may occur stochastically in time whenever a significant change in the input data environment occurs.

While the off-line optimisation approach is typically useful in dynamic problems that are computationally too complex for any algorithm to be applied in an on-line fashion, constructing and finding appropriate control laws can, however, be a complex task and is very problem specific. On the other hand, although much easier to implement than its off-line counterpart, the on-line optimisation procedure may significantly be handicapped by the static consideration of the problem during which the period of problem re-initialisation and optimisation procedures take place, as the theoretical model upon which the domain space is defined may no longer be a true and accurate representation of the present reality.

2.7.2 The notion of jump-start optimisation

It is often the case in many real-life dynamic problems that a certain level of urgency for finding high-quality alternatives to high-complexity problem instances at certain points in time is present. Although well adapted for static, once-off or sequential independent problem instances, the various metaheuristics discussed so far in this chapter ultimately fail to capitalise on the hidden benefits of domain space memory in the context of dynamic multiobjective optimisation problems. In other words, these search techniques are only configured to treat every new problem instance independently from the previous ones.

Unless changes in the domain space brought upon by input data alterations are very strong, it has been demonstrated that significant computational effort may often be saved, and better solution quality achieved, by reusing information gained in past problem instances when solving dynamic multiobjective optimisation problems [20, 21, 40, 56]. In particular, it may often be the case that any two consecutive feasible decision spaces in a dynamic multiobjective optimisation problem both contain identical subsets of solutions (provided that such solutions from the previous problem instance are evolved in time accordingly). More importantly, it may be the case that the intersection of any two consecutive Pareto optimal sets of solutions is significantly large.

The notion of *jump-start optimisation*[12] is the implementation of intelligent solution search

[12] For lack of a unified term pertaining to this concept in the literature.

techniques that consider (to varying extents) *a posteriori* knowledge on the nature of the decision and objective spaces configured during the *previous* problem instance at the start of the decision making process accompanying a *new* problem instance. Additionally, jump-start optimisation techniques may also be configured to track the shift in Pareto optimal solutions along the time continuum of a dynamic multiobjective optimisation problem.

2.7.3 Adaptive evolutionary mechanisms

It is reasonable to expect that evolutionary algorithms may prove to be useful in dynamic multiobjective problems, as they are inspired by a natural process that is subject to continuous adaptation. In particular, the occurrences of major input data changes on a high-quality, highly-diverse population of solutions in a dynamic multiobjective optimisation problem may, analogously, be compared to the effect that local and global events have on the individuals of a certain species (examples of such events include climatic changes, food sources depletion, natural cataclysms, diseases and predators upsurge), and how these individuals are able to cope and adapt to these events. Many studies therefore primarily focus on evolutionary techniques as a means of adapting a solution search process to dynamic multiobjective optimisation problems [12, 13, 20, 40, 74, 166].

In [40], for instance, various changes are performed on the original NSGA-II in order to adapt it to dynamic multiobjective optimisation problems. This adaptive evolutionary algorithm was renamed *DNSGA-II*. Due to a practical lack of consensus as to what level of input data variations are deemed "significant", this metaheuristic is not explicitly informed of all input data changes that take place. Instead, a sub-loop is required to test whether a significant change has occurred in the problem by selecting a few solutions at random from the current population and reevaluating them in both decision and objective space. If at least one change is observed amongst these random individuals, all parent solutions in the current population undergo full reevaluation, following which a new offspring population is generated. These two-subpopulations are then merged into a new, more representative population. Moreover, two versions of the DNSGA-II are presented in [40]. In the first version, a certain percentage of the newly adapted population is replaced with random immigrants, which supposedly works better whenever changes in decision and objective space are generally large. In the second version, instead of being replaced by random immigrants, the selected percentage of population members undergo mutation, so that the new solutions introduced into the new population are related to existing members of the population. It is claimed that this approach generally works better for dynamic problems in which small changes in input data generally take place.

According to Cámara *et al.* [21], an adaptive evolutionary optimisation tracking mechanism is required to continuously maintain diversity and generate uniformly distributed solutions along the non-dominated front as changes in the nature of the non-dominated objective space take place. In addition, solution transformation procedures (such as mutation and crossover) should ideally be set similarly to static problems, where an efficient trade-off is pursued between the exploration of solutions in new areas and the general convergence toward better solutions in the vicinity of the already good ones found.

In addition, Jin *et al.* [74] have stated in their research that *diversity after the changes* and *diversity along the runtime* are two very important notions that must be addressed in evolutionary dynamic multiobjective optimisation problems. On the one hand, the notion of diversity after the changes suggests that, as soon as a change in input data has been detected, diversity among population members should temporary be increased in order to access new regions of the search space and reassess the nature of the resulting new Pareto front. This is easily achievable using

a so-called *hypermutation* operator, where a sudden increment in the mutation rate is triggered over a fixed computational period as soon as an observable change in input data has taken place. On the other hand, the notion of diversity along the runtime suggests that close Pareto front convergence throughout the execution of the algorithm (particularly during the early stages of execution) should be avoided, so as to allow the algorithm to self-adapt and respond to the changes more effectively. This adaptation process may be achieved by inserting random immigrants in the population once a certain number of generations has occurred. An even more effective, yet more complicated, way to achieve this is to immigrate individuals exclusively to the newly available feasible subregions of the search space.

Moreover, according to Cobb and Grefenstette [28], there exist two basic strategies for modifying genetic algorithms to accommodate changing environments. The first strategy evolves around expanding the memory space of the algorithm in order to build up a repertoire of ready responses for pre-defined environmental conditions. The second strategy, similarly to that followed in [74], is to employ some method for increasing diversity in the population (such as hypermutation) as a means of compensating for changes encountered in the environment.

Finally, in the process of seeking non-dominated fronts throughout their adaptive evolutionary algorithm, Farina *et al.* [56] have suggested that there exist four fundamental ways in which a problem can demonstrate time-varying changes, namely (1) changes in Pareto optimal decision variables ($\mathcal{P}$) but not in the objective function mappings of Pareto optimal solutions ($\mathcal{F}$), (2) changes in both $\mathcal{P}$ and $\mathcal{F}$, (3) changes in $\mathcal{P}$ but no changes in $\mathcal{F}$ and (4) no changes in either $\mathcal{P}$ or $\mathcal{F}$.

2.8 MLE DSSs in existence

Complex decisions made by system operators are usually supported to some extent by certain automated assistance tools, better known in the literature as *decision support systems* (DSSs). These tools aim to provide the operators with a set of high-quality alternatives in a limited time for solving specific problem instances, particularly when dealing with decisions that are complex and which are subject to a high level of uncertainty. This output may then be used in conjunction with operator judgment (based on extensive knowledge, experience in the field and training) so as to select a single, most preferred alternative for solving the problem at hand. A DSS does not usually have the power to override operator decisions; the operator normally has the final say in the outcome of the decision making process, although in extreme cases, the operator may choose to implement one of the alternatives proposed by the DSS without contributing any personal input to the decision process himself. On the other hand, if the problem complexity is relatively low, the operator may simply prefer to make a subjective decision regardless of the information provided by the DSS. In any case, alternatives proposed by a DSS should not surprise an experienced operator, but rather confirm what he intuitively believes to be good solutions to the problem at hand, as well as ease his conscience were he to experience some form of doubt.

It is instructive to review some of the material presented in previous MLE-related research in order to gain an understanding of certain aspects relevant to the scope of this dissertation. Information related to the detection, evaluation and tracking of VOIs at sea provides most of the essential input data for the MLE response selection process. These input data play a critical role in the design of models for assessing the potentially threatening behaviour of VOIs at sea and, as a result, many studies have been devoted to the processes of data collection, and understanding and documenting the processes of threat detection, threat evaluation and vessel

tracking [9, 49, 60, 62, 82, 85, 102, 116, 125, 158]. These publications often contain work by groups of researchers from different nationalities who, together, possess a large set of diverse skills and accumulated experience with respect to the design of high-performance DSSs.

2.8.1 MLE threat detection DSSs

The *threat detection* process should provide an MLE response selection operator with kinematic information, such as the locations and velocities, of potentially threatening VOIs at sea, which are, in turn, required for the assignment decision process underlying the MLE response selection process.

The *Integrated Maritime Surveillance System* described by Ponsford *et al.* [116] is an example of a shore-based system which detects, tracks, classifies and identifies surface and air targets within the EEZ of a coastal nation. In general, the detection process is carried out using high-performance radars[13].

In [62], a so-called *Automated Identification System* is described, which is a large detection infrastructure including infrared cameras, an airborne platform carrying a radar, and various video cameras. The overall functions of this automated decision support system, however, encompass the detection, tracking, identification and classification of multiple targets, as well as the evaluation of their respective levels of threat and the selection of a course of intervention with respect to these threats (*i.e.* MLE response selection for the interception of these targets).

2.8.2 MLE threat evaluation DSSs

In contrast to the threat detection process, in which the focus falls on being aware of the presence of potentially threatening VOI locations in real time, the *threat evaluation* process should provide the operator with expectations in respect of the nature of a potentially threatening event at sea, the degree of "oddness" associated with its behaviour (such as deviation from standard trajectories, suspect encounters between vessels at sea, quick in-and-out crossing of the EEZ boundary or rapid pursuit of a commercial vessel by an unidentified vessel) as well as expectations with respect to the levels of threats that these VOIs may induce.

Farina *et al.* [55], for example, categorise VOIs as being either "neutral," "suspect" or "threatening" for objects heading towards a coastal-based structure. They mention that threat evaluation depends on the target type, and is typically based on a deterministic comparison between the target kinematic parameters (namely speed, distance from the coast and course) and certain allowable values defined by tolerance thresholds.

In [116], part of the system includes the use of a broad range of sensors which provide target information on an *ad-hoc* basis to distinguish non-cooperative from cooperative targets. The basic idea behind their threat evaluation logic is to delay classification of threat levels when not enough information is available to determine the nature of the object.

Furthermore, the innovative work of Dabrowski and de Villiers [36] contains a framework for modelling maritime vessel behaviour, and in particular piracy situations, using simulated track data. This framework is based on a dynamic Bayesian network modelling paradigm and is applied to generate a hypothetical data set of pirate locations. The framework is evaluated by comparing this data set to a real-world data set consisting of locations of pirate attacks that have

[13] A radar's performance is typically measured by its ability to detect and track targets at long ranges, and to resolve targets in close proximity to one another [116].

occurred over the course of a year by employing cross validation techniques. It is demonstrated that the framework has the ability to simulate data that are similar to real-world data sets. A meta-framework for evaluating behavioural models is also introduced in [36].

2.8.3 MLE resource assignment DSSs

The decision making process involved in MLE response selection by developing coastal nations is often exclusively conducted on the basis of human judgment, without the use of automated DSSs [49]. These decisions are usually made by one or more individuals, based on intuition and (often considerable) experience in the MLE domain, in contrast to being the result of an automated, analytical process. Although many technical MLE-related papers have been published in the past, relatively few primarily focus on resource assignment problems and, of those, most primarily focus on the management of non-active (*i.e.* idle) resources[14], rather than on assigning resources to missions for intercepting VOIs in real time in a generic manner [38, 54, 91, 103, 129, 158]. A small niche of studies nevertheless consider the MLE response selection process in significant detail. For example, in order to address the increase in pirate attacks against shipping and oil field installations, Bouejla *et al.* [9] propose an innovative solution search process to counter the problem of offshore piracy from the perspective of the entire MLE flow of events (*i.e.* from the detection of a VOI to the implementation of a response). According to this study, the response selection process must take into account multiple variables, such as the characteristics of the threat, the potential target, the available protection resources, as well as various environmental constraints. The use of Bayesian networks is employed as a means of managing the large number of parameters associated with this process and identifying appropriate counter-threat measures. The contents of two of the impact papers referred to above are summarised in some detail in the remainder of this section.

Dynamic configuration and management of multiple MLE resources

Farahbod *et al.* [54] present an extensive *Dynamic Resource Configuration & Management Architecture* DSS. The aim of this DSS is to address adaptability requirements with respect to dynamic changes in the availability of mobile resources and the service they provide for surveillance and rescue missions.

Here, the underlying architecture adopts a decentralised organisational structure which deals with internal changes in resource requirements and external changes impacting on the availability of resources. A hierarchical network of nodes representing mobile resources is assumed in this structure. The reason for a decentralised approach is motivated by the scale of the system to be managed, particularly with respect to the very large volume of input data and information to be processed, the complexity of tasks and the dynamic nature of mission implementations.

Two fundamental components are considered in the design of this DSS, which are responsible for controlling the logical and physical distribution of resources in both time and space, namely *resource assignment* and *task management*. Resource assignment involves the orchestration of the operational resources, such as resource clustering, migration policies and monitoring capability changes. Task management, on the other hand, deals with complexities associated with missions and their decomposition into tasks to be performed by available resources, such as task allocation, task prioritisation and task scheduling.

[14] A clear differentiation between active and idle MLE resources will be made at a later stage in this dissertation.

In line with the task management component, a model based on the concept of a *task lifecycle* is presented, which is defined as an abstract operational framework for controlling and managing the resource processing tasks in a local environment. The execution of these tasks is continued over a series of temporal stages.

In line with the resource assignment component, two types of resource distribution processes are identified, namely *physical* and *logical* distribution. Physical distribution involves the spatio-temporal distribution of resources in a geographical environment. Logical distribution, on the other hand, refers to the dynamic configuration of physical resources into clusters, which vary according to changes in task management procedures, changes in resource capabilities (*e.g.* breakdowns or previously unavailable resources becoming available) and changes in the environment (*e.g.* weather conditions and oceanic currents). More specifically, certain resources may be clustered into a higher resource level associated with augmented capabilities required to perform more complex tasks. These clusters are configured to operate in a semi-autonomous manner and to reduce control and communication costs by making local decisions in respect of the task allocation of individual resources within the cluster. Overall, these two distribution processes are required, *inter alia*, to maintain a desirable assignment workload balance within and between the clusters of the network.

The cutter scheduling problem

Darby-Dowman *et al.* [38] present an automated DSS for producing a schedule for a fleet of ships on a daily basis, over a one-year time frame. It is used for tactical and strategic purposes as an investigative tool for the US Coast Guard.

The proposed DSS is fully integrated with a mathematical model responsible for schedule generation and optimisation. The operator is, however, able to make decisions without any specialised mathematical knowledge. The purpose of the DSS is not to replace the operator, but rather to provide him with a powerful automated tool to significantly enhance his decision making effectiveness. It is stated in [38] that decision problems may be classified as either *programmed* or *non-programmed*. Programmed decisions are characterised as structured, routine and repetitive. They can be delegated to lower echelons of the organisation and do not require intensive supervision. Non-programmed decisions, on the other hand, are characterised as unstructured, complex and abstract; they cannot be established by well-defined procedures and solving them typically requires creativity, subjective input and abstract thinking. Consequently, senior managers usually deal with decisions of this type due to their considerable experience in the field.

The various missions undertaken by the US Coast Guard in its maritime jurisdiction areas consist of search and rescue, law enforcement, response to maritime environmental incidents, illegal and regulated fishing, customs regulation, and vessel safety. The jurisdiction area around the US is partitioned into *district areas*. Each of these areas is managed by a *district commander*, responsible for overseeing the operational and administrative matters occurring in his district. In addition, an *area commander* is responsible for overseeing situations that concern multiple district areas with regard to the geographical region, the scope or the resources.

Operations at sea are carried out by a heterogeneous fleet of vessels and aircraft, defined in [38] as *cutters*. The primary task of these cutters evolves around patrolling in a number of maritime areas until needed for a counter-threat or reconnaissance mission, but they are also used for other types of assignments, such as planned maintenance, attendance at public events and training exercises. Each cutter belongs to a certain class of MLE resources with specific performance characteristics. Additionally, ports with unique attributes are located along the coastline for the purpose of undertaking various types of maintenance and spending allowable

transit times between patrol schedules. Each cutter is associated with a single port and thereby inherits the specified attributes of that port.

In this DSS, a mathematical modelling and optimisation subsystem is responsible for solving the *cutter scheduling problem*, which involves the optimal allocation of cutters within time windows in a dynamic environment while dealing with various subjective and objective issues. A major challenge faced in this process is that these schedules require real-time revision to account for unplanned maintenance, military actions or any other particular situations that are external to the model.

An integer goal programming formulation in the form of a set partitioning model is adopted to model the problem. Two requirements placed on the cutter scheduling process are accounted for in the model. One deals with minimising the level of coverage of cutters, and the other deals with regular training and cutter maintenance. In addition, upper bounds are placed on the number of cutters undertaking these tasks at any given time. Because MLE roleplayer-induced requirements may, however, often result in the non-existence of feasible solutions to such a problem instance, a relaxation of these requirements is also conducted using penalising functions. The solution "requirements" may therefore be interpreted as desirable attributes, rather than strict, inflexible conditions. A branch-and-bound search strategy is then employed in order to find a near-optimal integer feasible solution to a given problem instance.

Finally, the DSS is also enhanced with a post-solution analysis subsystem responsible for inspecting and diagnosing the schedules generated. Using a list of rules which has been provided by experienced schedulers, this subsystem informs the operator of potential functional problems for a particular cutter schedule. This is motivated by the idea that there exist many situations which require human intervention after the generation of a cutter schedule, as the mathematical model cannot account for all possible scenarios.

2.9 Chapter Summary

The purpose of this chapter was to provide the reader with sufficient information from the literature so as to facilitate an understanding of the concepts presented later in this dissertation. In §2.1, a discussion was conducted on the philosophy and fundamental concepts behind decision making, and this was followed by an introduction to decision making problems with multiple conflicting objectives. A mathematical description of the notion of dominance and Pareto optimality was then provided and the section closed with a discussion on multiperson decision making. In §2.2, a general discussion on search techniques was conducted, and a method of classifying these techniques was presented. The popular use of metaheuristics for solving complex multiobjective optimisation problems in the literature was also motivated in this section. Following this, the nature and working of simulated annealing and evolutionary algorithms were outlined and discussed thoroughly in §2.3 and §2.4, respectively, within the context of both single and multiobjective optimisation problems. In addition, the methods of tabu search, ant colony optimisation and particle swarm optimisation were described in §2.5 within the context of single objective optimisation problems. A brief review of the classical VRP and a number of its variants was then presented in §2.6. The nature of the contents in this section focused particularly on the role of the nature of information with respect to defining, modelling and solving difficult VRP variants, the principles behind dynamic and stochastic VRPs, as well as popular methodologies employed in the literature for solving multiobjective VRPs. In §2.7, various intricate concepts and technical approaches considered in the context of solving dynamic multiobjective optimisation problems in general were discussed. Finally, a review of MLE DSSs

in existence was conducted in §2.8, with an emphasis on MLE resource assignment DSSs, where two high-quality relatable studies were summarised in some detail.

Based on most of the reviewed MLE DSSs in existence, it appears that the MLE response selection processes followed by a coastal nation are typically informed by five major factors: the set of MLE resources available, the geographical layout of the jurisdiction waters, the priorities assigned to neutralising various threat types in the jurisdiction waters, the assignment of idle MLE resources to bases or patrol circuits, and the opinions of multiple role players and decision makers regarding the courses of action to be followed in an MLE environment. Very few studies, however, tend to focus on the MLE response selection process for active MLE resources in real time, and no previous studies were found in the literature treating the MLE response selection problem as a combinatorial optimisation problem, adopting the analogy of the dynamic VRP as a means to model the problem. In addition, most studies are usually conducted within the context of the environment of a specific coastal nation and thus tend to not be formulated in a generic manner. In the following chapter, therefore, a generic architecture is presented for a generic MLE response selection DSS within three generic autonomy paradigms of decision making.

CHAPTER 3

System Architecture

Contents

In §2.7, a literature review of existing MLE DSSs was conducted and it was observed that such DSSs tend to be designed for use in very specific scenarios, for use by a certain coastal nation only or without primarily focusing on the MLE response selection process as defined in Chapter 1. A generic system architecture is therefore proposed in this chapter for use in coordinating the MLE operations of a coastal nation. The architecture is discussed in some detail in §3.1 in order to capture the essence of the process involved and this is followed by a description of the chronological order of MLE events running concurrently within the context of this architecture in §3.2. A discussion of three hypothetical paradigms with respect to the autonomy of decision makers using the DSS is then conducted in §3.3. Finally, the chapter closes in §3.4 with a brief summary.

3.1 Functional elements in an MLE environment

As discussed in Chapter 1, decisions in a typical MLE environment may be classified into three broad fields, namely decisions related to threat detection, threat evaluation and resource assignment. The MLE system architecture proposed here consists of ten classes of functional elements, namely *external MLE threat detection infrastructure* elements, *external MLE threat evaluation infrastructure* elements, *external MLE resource assignment infrastructure* elements, an *attribute management* system, a *VOI flagging DSS*, a *VOI threat analysis DSS*, an *MLE response selection DSS*, an *idle MLE resources management DSS*, an *MLE Human Machine Interface* (HMI), and

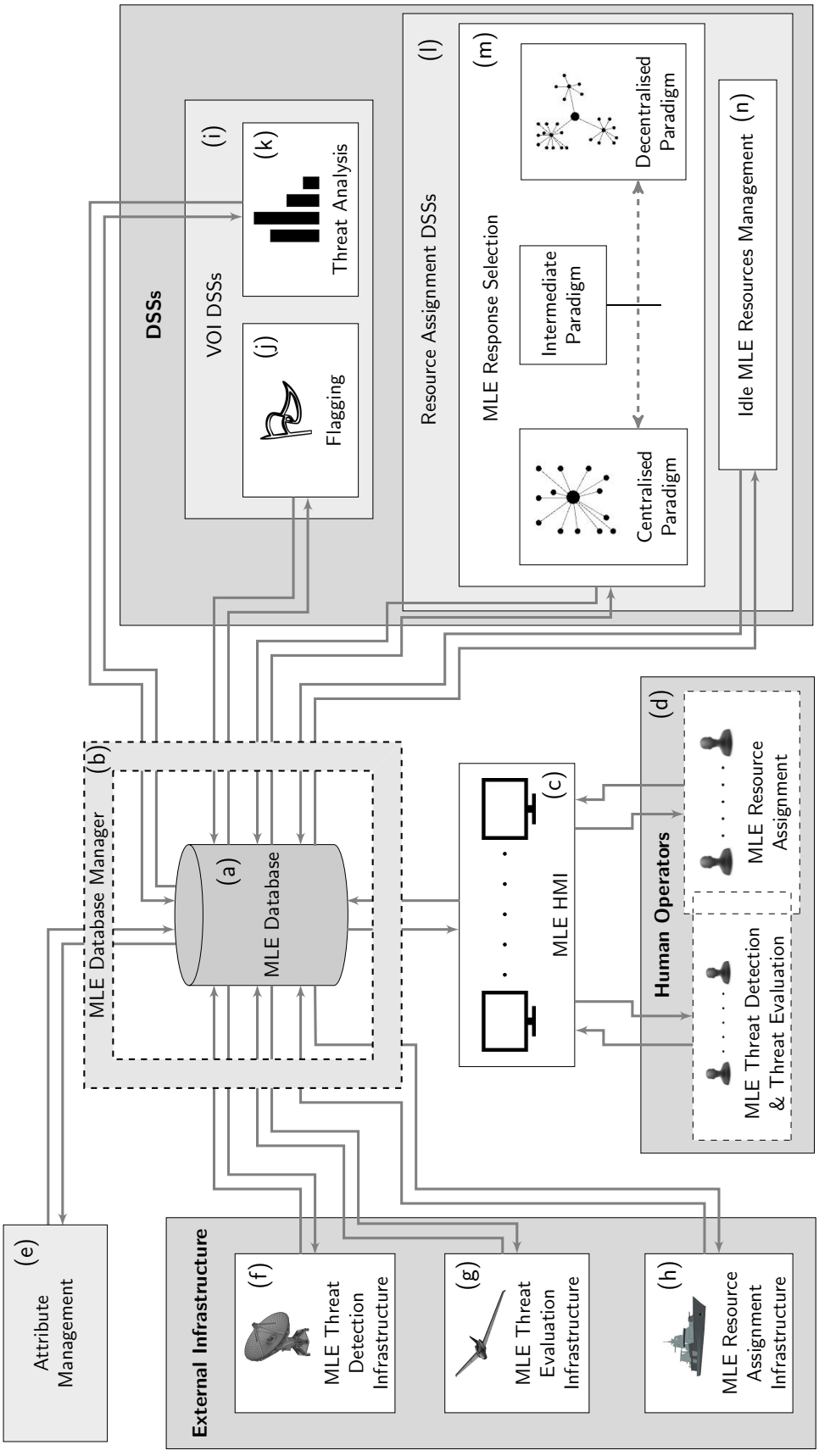


FIGURE 3.1: *MLE functional elements and the flow of information between these elements.*

an *MLE database* and associated *database manager*. These functional elements, together with the flow of information between them, are illustrated in Figure 3.1.

3.1.1 Overview of the proposed architecture

The first and most central functional element in Figure 3.1 is the MLE database manager and its associated central database, denoted in the figure by (a) and (b), respectively. This database is a storage and retrieval mechanism housing a large collection of data, and it also serves as the communication hub of the system from where data are continually transmitted between the functional elements of the system — all functional elements therefore communicate with one another *via* this database. The associated database manager is responsible for managing updates to or facilitating queries for MLE operations-related data as well as processing and converting output data from one functional element to the correct input data format for another.

The HMI in Figure 3.1(c) is the link between man and machine; it enables a human operator to communicate with the functional elements of the system by providing him with an interactive graphic user interface which displays the situation at sea in real time together with the results generated by the other functional elements on a series of computer screens. It is advocated that this transfer of information should be implemented in the form of so-called *data streams*, which are strings of bits associated with *identifiers* that allow an operator to request specific data in real time. These streams may be fed live to the HMI and, depending on whether or not they are activated by the operator, may be visualised on a display screen. For instance,

$$[\textit{VesselIdentifier; (Long,Lat); Course; Speed}] = [0020; (-35.83, 18.90); 120.3; 18]$$

is an example of a snapshot from a data stream which may be visualised in a user-friendly fashion *via* the HMI for the benefit of a threat detection operator, while

$$[\textit{VesselIdentifier; Status; FlagCode}] = [0113; Flagged; ZoneInfraction]$$

is an example of a data stream snapshot which may be displayed similarly *via* the HMI for the benefit of a threat evaluation operator. Here, the "VesselIdentifier" data field is used as a unique identifier for attaching information to a particular VOI.

In this system architecture, it is proposed that, as the MLE threat detection and threat evaluation processes tend naturally to function closely together, so too should the MLE resource assignment operators in Figure 3.1(d) work closely together with the MLE threat detection and threat evaluation operators. In certain cases, however, where there are a very limited number of MLE human resource operators, or if a certain human operator has considerable expertise in all three MLE areas, one or a small number of operators may very well be responsible for decision making in all aspects of MLE.

The external threat detection infrastructure functional elements in Figure 3.1(f) comprise all devices used, directly or indirectly, for detecting and tracking maritime objects in specified areas of the ocean as well as for gathering kinematic information associated with these objects, such as their sizes, locations and velocities. Examples of such threat detection devices are land-based, aerial or vessel-based radars, infrared cameras, video surveillance cameras and observations reported from active naval, aerial or reconnaissance MLE resources.

These measured attributes of VOIs, obtained from the sensor systems, are fused together by the attribute management system in Figure 3.1(e) to create so-called *system tracks* and *derived attributes* associated with each of the VOIs within the jurisdiction area of the coastal nation [49].

A system track is a record of the historical displacement of a VOI in space from its detection to the present time, while a derived attribute of a VOI is a parameter value that is computed from the measured attributes of the VOI, such as its acceleration or rate of changing course. Based on these attributes, the system also provides future VOI trajectory estimates to the DSSs. Track stitching methods and Markov models are examples of techniques that may be employed to assemble system tracks [49].

This fusion process is conducted by employing all data and knowledge associated with a VOI and results in a combination of these elements into a single consistent, accurate, and useful data representation for that VOI. Successfully implementing the fusion process is critical in cases where VOI input data are obtained from multiple threat detection infrastructure sources which may individually yield slightly or significantly different perspectives on information. The *JDL data fusion model*, for example, is perhaps the most widely-used method for categorizing data fusion-related functions [2]. This model has been shown to be useful across multiple application areas and is typically intended to identify the processes, functions, categories of techniques, and specific techniques applicable to data fusion.

In order to filter VOIs from the multitude of maritime objects tracked at sea within the jurisdiction area of a coastal nation, threat detection operators are tasked with analysing these objects in respect of the physical and kinematic information attached to them, and assessing whether their behaviour is deemed unexplained, suspicious or threatening (to some extent). Following this analysis, the status of a maritime object may be designated as either a VOI or a maritime object of no interest. This procedure must be carried out on a continual basis as the situation at sea unfolds, typically involves a large number of maritime objects and may therefore easily overwhelm a human operator. The aim of the VOI flagging functional element in Figure 3.1(j) is to assist human operators in their threat detection decisions by providing automated decision support for identifying and tracking suspicious maritime activities.

An adaptive early-warning maritime surveillance DSS capable of supporting a maritime operator in his decision making process with regard to various threats and scenarios was proposed by du Toit [49]. Here, so-called *fusion*, *rule-based system*, *activity classifier* and *data mining* components are incorporated into the proposed DSS. Moreover, the HMI component in this DSS provides the operator with relevant information as well as a mechanism by which the operator may provide feedback to the system. The fusion component, for instance, processes the information received from threat detection devices, such as radars or infrared cameras, in the form of vessel track updates, which are stored in an operational database. This operational database is directly utilised by the rule-based system component which processes the data as they become available and aims to identify potentially threatening activities amongst the observed vessel tracks according to certain pre-configured sets of rules, such as rules pertaining to zone infractions, proximity of vessels or anomalous activities. Data from the operational database are migrated to a central database after being cleaned and transformed into the necessary format. As tracks become available in the central database, they are processed by the activity classifier component. The work in [49] is strictly limited to the maritime domain and the primary source of data is spatio-temporal AIS data.

The input data provided by the external threat detection infrastructure functional elements may not always provide all the information necessary to facilitate good inferences with respect to the potentially threatening nature of a VOI. The external MLE threat evaluation infrastructure functional elements in Figure 3.1(g) comprise a collection of physical devices which may be used, when requested by an operator, to collect more detailed and/or accurate information related to a particular VOI. Scouting resources (such as unmanned airplanes) dispatched to specific maritime zones in order to establish line-of-sight contact with respect to targeted VOIs,

and advanced radar systems whose primary function is effectively to zoom in on particular VOI locations, are examples of such devices.

The purpose of the VOI threat analysis DSS in Figure 3.1(i) is to assist a threat evaluation operator in classifying and quantifying the potentially threatening nature of VOIs, by providing automated decision support for analysing the behaviour of VOIs based on the information collected during the threat detection process and making automated inferences with respect to the nature and level of threat posed by VOIs. This process therefore only involves system tracks of VOIs and not the tracks of maritime objects deemed of little or no interest. Ultimately, the output of such a DSS consists of a vector associated with each VOI containing entries which correspond to estimated probabilities with which the VOI in question belongs to a finite number of pre-determined threat classes, an unknown threat class and a false alarm class. Various mathematical models may underlie this process of quantifying these probabilities. Bayes' theorem may, for example, be used as a framework for introducing newly observed information [9, 36].

The MLE resource assignment subsystem in Figure 3.1(l) is responsible for assisting human operators with decisions related to the utilisation of MLE resources in both time and space. These MLE resources are generally either allocated for the purpose of intercepting VOIs at sea (such resources are said to be in an *active* state) or are otherwise strategically allocated to certain patrol areas at sea or moved to bases until needed for future law enforcement purposes (such resources are said to be in an *idle* state). Whereas the MLE response selection DSS in Figure 3.1(m) deals with the allocation of active MLE resources, the idle MLE resources management in DSS Figure 3.1(n) deals with the allocation of idle MLE resources. These two functional elements should operate in perfect real-time synchronisation with one another.

Tasks performed within the idle MLE resources management DSS involve, *inter alia*, the configuration of patrol schedules, the planning of repair and maintenance of resources and the design of effective scouting strategies. This DSS should combine the expertise and experience of human operators with large amounts of historical data (*e.g.* probability distributions of specific threatening activities occurring in specific zones of the jurisdiction area over specific time intervals) and various mathematical models for scheduling idle MLE resources in both time and space. Because occurrences of newly detected VOIs are stochastically distributed in space and time, it is crucial to manage idle MLE resources in such a way that leaves them, *a posteriori*, in central positions of relative readiness so that they may be dispatched rapidly when shifted into an active state.

3.1.2 External MLE resource assignment infrastructure

As discussed earlier, all physical infrastructure elements dealing with the assignment of MLE resources reside in the external MLE resource assignment infrastructure functional element class in Figure 3.1(h). Five major components that are relevant to MLE response selection operations have been identified. These are a geographical information system model of the jurisdiction area, used to define clear boundaries at sea and to incorporate rules and regulations that are applicable within these boundaries; a set of decision making entities comprising the various MLE resource assignment role players of a coastal nation; an MLE fleet of resources comprising one or more classes of units, each class possessing a clear, unique set of attributes and capabilities as well as a parent decision entity; a set of MLE resource assignment bases providing various types of essential services to idle MLE resources; and a set of pre-defined circuits used for patrol purposes. These five components are described in more detail in this section.

The jurisdiction area

The set of the maritime zones, as defined in the UNCLOS [151], should be used as a guideline to define the maritime law enforcement boundaries of a coastal nation. These boundaries should therefore at the very least enclose the territorial seas, the contiguous zone and the exclusive economic zone as jurisdiction areas, but these areas may be subdivided or refined further, as required by the coastal nation in question in order to meet specific MLE goals. Farina *et al.* [55], for example, simulate a maritime border control environment with an off-limit zone and a warning zone, measured at 20 km and 50 km from the coastal baseline, respectively. The jurisdiction area of a coastal nation may be modelled using shape-lines generated by standard Geographic Information System software packages.

The decision entities

As mentioned in §1.2, it is typically the case that the entire MLE response selection process of a coastal nation is not conducted by a centralised operator assisted by a single DSS, but is rather orchestrated by multiple role players, called MLE *decision entities*. These decision entities may perceive the quality of MLE response selection operations in their own, different ways, as they each tend to pursue their own goals and subjective perceptions of what is deemed important while carrying out MLE operations. In particular, these decision entities may perceive the threatening intensity of VOIs differently, function largely independently from one another, follow their own guidelines, and utilise their own subsets of MLE resources.

The South African Navy, the South African Maritime Safety Authority (SAMSA) and the Department of Agriculture, Forestry and Fisheries (DAFF) are three examples of major decision entities in the South African MLE environment [49]. While the South African Navy specialises in investigating and neutralising acts of piracy, threats to sovereignty[1] and terrorism threats, SAMSA specialises in investigating and neutralising incidents of pollution, accidents at sea and hazardous vessel threats, while DAFF resources specialise in investigating and neutralising illegal fishing threats. Sometimes these decision entities function independently and sometimes they collaborate. For example, the South African Navy and SAMSA often combine their resources when investigating and neutralising smuggling and illegal immigration threats.

The MLE resources

MLE operations require law enforcement resources capable of neutralizing a variety of threats at sea. These resources perform differently, depending on their physical characteristics (such as size, maximum speed, manpower on board, weapon system infrastructure and other defense mechanisms, autonomy at sea, and set-up time). Here, the speed of a resource influences its ability to intercept VOIs effectively, while its weapon system infrastructure and other defense mechanisms determine its effectiveness with respect to neutralising threats. The autonomy of a resource refers to its ability to be self-sufficient at sea for an extended period of time while on a mission, whilst its set-up time refers to the time required to prepare the resource for departure on a mission.

The optimal resource fleet composition of MLE resources is not addressed as part of this MLE resource assignment problem; the set of MLE resources at the disposal of a coastal nation is normally assumed fixed and, consequently, the fixed costs associated with the acquisition of MLE

[1] A foreign warship arriving unannounced in the jurisdiction area of a coastal nation constitutes a threat to the sovereignty of the coastal nation.

resources are external to the problem. The variable (set-up and operating) costs associated with these resources are, however, taken into consideration.

The simulator proposed by Farina *et al.* [55], for example, incorporates two types of resources, namely a helicopter and a patrol boat. Here, the resource response time is calculated based on the velocity of the target and the resource parameters. The system considers the current MLE resources available in order to select an appropriate resource to be dispatched to intercept the target. The resource parameters considered in [55] are availability, speed, inspection time and departure time.

The MLE resource bases

A base may be defined as a facility providing the necessary re-supplying, stationing and maintenance services to MLE resources. Every base is typically unique with respect to certain characteristics that are relevant to resource assignment operations, such as size, accessibility or geographical location. These characteristics, combined with external factors, may have a significant impact on the way MLE resource assignment operations are conducted. Bases may be classified as either *inland*, *coastal* or *deployable*. Inland bases are usually accessible to only certain types of MLE resources if connected to the maritime jurisdiction area *via* a network of rivers, or only accessible to aerial MLE resources otherwise. Coastal bases are located at fixed locations along a coastline and provide a vast range of services to a variety of MLE resource types. Finally, deployable bases have the ability to be dispatched in both time and space, from where they, themselves, may dispatch fixed sets of MLE resources associated with the particular bases.

The work of Malik *et al.* [91], for example, deals with the optimal allocation of MLE resources distributed amongst three stations (bases) to undertake assignments in the Great Lakes region of North America. The study focused on determining the spatial and temporal distribution of response cases and their assorted MLE resources for all search and rescue operations and how closing any one of these three stations may affect the workload of the other stations that would subsequently be required to absorb the additional tasks. It was concluded that the closure of auxiliary stations would imply longer response times, which may potentially translate into the loss of lives and property. Overall, the DSS in place allowed users to conduct a sensitivity analysis by determining the change in risks associated with closing certain stations in terms of the success rate of search and rescue operations in the medium and long run.

The patrol circuits

In an MLE environment, the notion of *patrol* may be described as the strategic allocation of idle MLE resources in specific regions of the jurisdiction area. A *patrol circuit* may be any pre-defined maritime zone boundary, route or singular point to which an idle MLE resources may be assigned strategically. A patrolling MLE resource typically breaks away from its designated patrol circuit as a result of one of the following occurrences: (a) the MLE resource has travelled further than a certain maximum distance threshold and is re-allocated to performing a different task, (b) the MLE resource has travelled for more than a pre-determined maximum amount of time since it was last re-supplied and is therefore re-allocated to performing a different task, (c) the MLE resource is assigned a mission to intercept one or more VOIs (*i.e.* the MLE resource transits from an idle state to an active state) or (d) an operator decides to re-allocate the MLE resource to a different patrol circuit or to a base for any reasons other than the occurrences (a) and (b) above.

Shieh *et al.* [129], for example, introduce an interesting game theoretic DSS premised on an attacker-defender Stackelberg game model, where the adversary is defined as an individual with ties to al-Qa'ida. This DSS has been deployed by the US Coast Guard to schedule patrols around the port of Boston. The DSS is currently in the process of being deployed around the port of New York as well.

3.1.3 The MLE response selection DSS

The proposed core MLE response selection DSS consists of a *model generation* subsystem, a *model solution* subsystem and an *information management* subsystem. The model generation subsystem comprises a *model configuration* component, which allows for the construction and storage of fundamental mathematical structures and modelling components (such as objective functions, routing constraints and MLE resource parameters), and a *model management* component, which accommodates dynamic features that are used to model the problem on a temporal basis. The solution search subsystem is concerned with finding and presenting a set of non-dominated trade-off solutions to the MLE response selection operator for every problem instance formulated in the mathematical modelling subsystem. The first component of the information management subsystem is the *problem instance reinitialisation* component, which is responsible for detecting significant changes in input data transferred from relevant external sources, and to inform the operator of the extent and nature of these changes. The second component is the *optimisation tracker* component, which acts as an independent, automated solution search engine; its purpose is to track the (possibly approximate) Pareto-optimal set of solutions in real time while considering all input data changes taking place. The last component is the *solution tracker* component. Here, solution tracking may be thought of as an independent process which continuously updates, filters and stores strategic solutions to the current real-time situation for eventual use in MLE response selection-related decisions. This component is also responsible for communicating minor changes observed in the currently implemented solution to micro-management role players, following which the necessary adjustments in solution implementation can be made. These subsystems share input data *via* the centralised MLE database, as illustrated schematically in Figure 3.2. Here, the *external data sources* node refers to input data received from functional elements that are external to this DSS (see Figure 3.1).

The overall working of the proposed MLE response selection process is summarised as follows. A new cycle of the DSS, or *problem instance*, is initiated whenever the problem instance reinitialisation component detects a significant change in input data. This event in time is called a *disturbance*. Disturbances may cause the currently implemented solution no longer to be considered the most preferred MLE response selection alternative, as these are typically likely to cause current solution changes on a macroscopic level. The rate of reinitialisation of problem instances over time is therefore proportional to the rate of flow of events triggering disturbances. Before the next problem instance may be initiated, however, it is advocated that the operator be required to confirm that the detected changes in input data are indeed categorised as a disturbance.

Following the advent of a disturbance, the new MLE situation is remodelled as a new MLE response selection problem instance in the model configuration component. In addition, the operator should be able to participate in the process by contributing input based on his expertise in the field in respect of certain subjective conditions or preferences related to the current MLE situation *via* the model management component.

Following the mathematical modelling process pertaining to a problem instance, the solution search subsystem is required to find a high-quality, filtered set of so-called non-dominated solu-

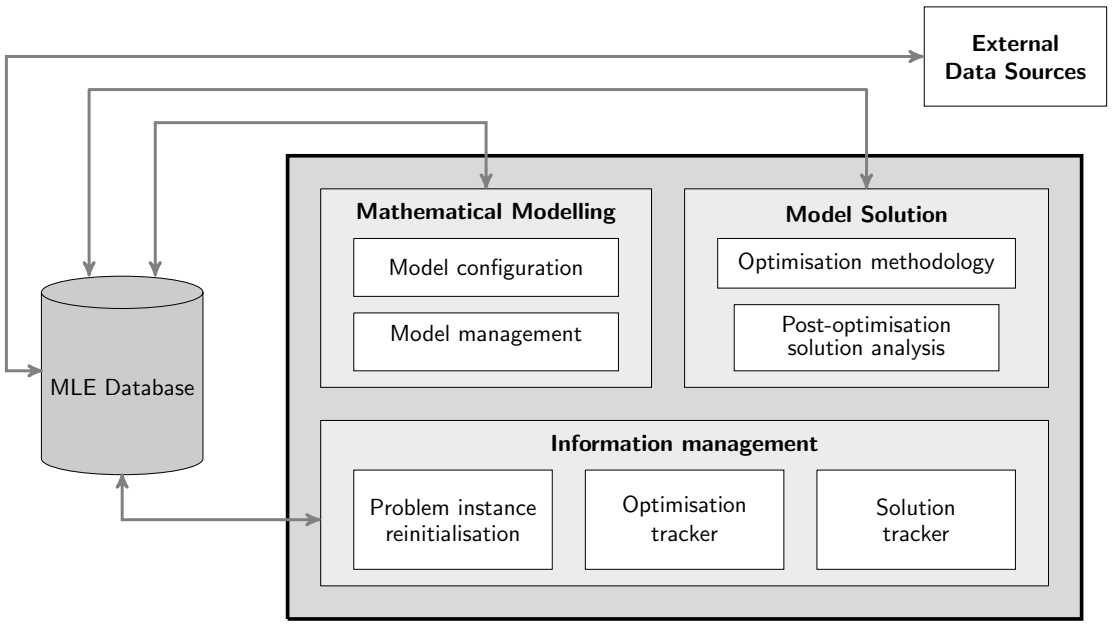


FIGURE 3.2: *The MLE response selection DSS with its three subsystems and their components.*

tions in multiobjective space using effective (metaheuristic) search techniques. Because the time elapsed between the previously implemented solution and the new one causes a deterioration in the effectiveness of MLE response selection operations, it is critical for these search techniques to be able to find high-quality solutions to the new problem instance in a very limited time.

The operator is then required to select a single (most preferred) alternative from this set of suggested trade-off solutions in the post-optimisation solution analysis component, which consists in assigning a subset of MLE resources to intercept and investigate a subset of VOIs. This solution should finally be communicated to the MLE database, from where the various MLE response selection decision entities and other various role-players may immediately be notified of the changes in routing implementations brought along by this new solution.

Meanwhile, the optimisation tracker and solution tracker components should operate in parallel along a time continuum, tracking a high-quality, highly diverse set of non-dominates alternatives and managing these alternatives as the MLE response selection situation changes in real time and new input data are made available. These solutions should be made available to the mathematical modelling and model solution subsystems at the start of a problem instance as a means of speeding up the decision making process brought along by the start of a new problem instance.

3.2 The chronological order of MLE events

The functional elements described in the previous section operate as part of a so-called *MLE cycle* of events, inspired by the so-called *Threat Evaluation Weapon Assignment cycle* proposed by Lötter & van Vuuren [89] in the context of ground-based air defense. A trigger of such a cycle initiates consecutive calls to these elements in an orderly fashion so as to provide updated data to the operators in real time. It is assumed that an MLE cycle is triggered by a significant change in any of the input data fields of the MLE database or by the natural continuation of an implementation clock cycle. The sequential order in which the functional elements are utilised within an MLE cycle is illustrated graphically in Figure 3.3.

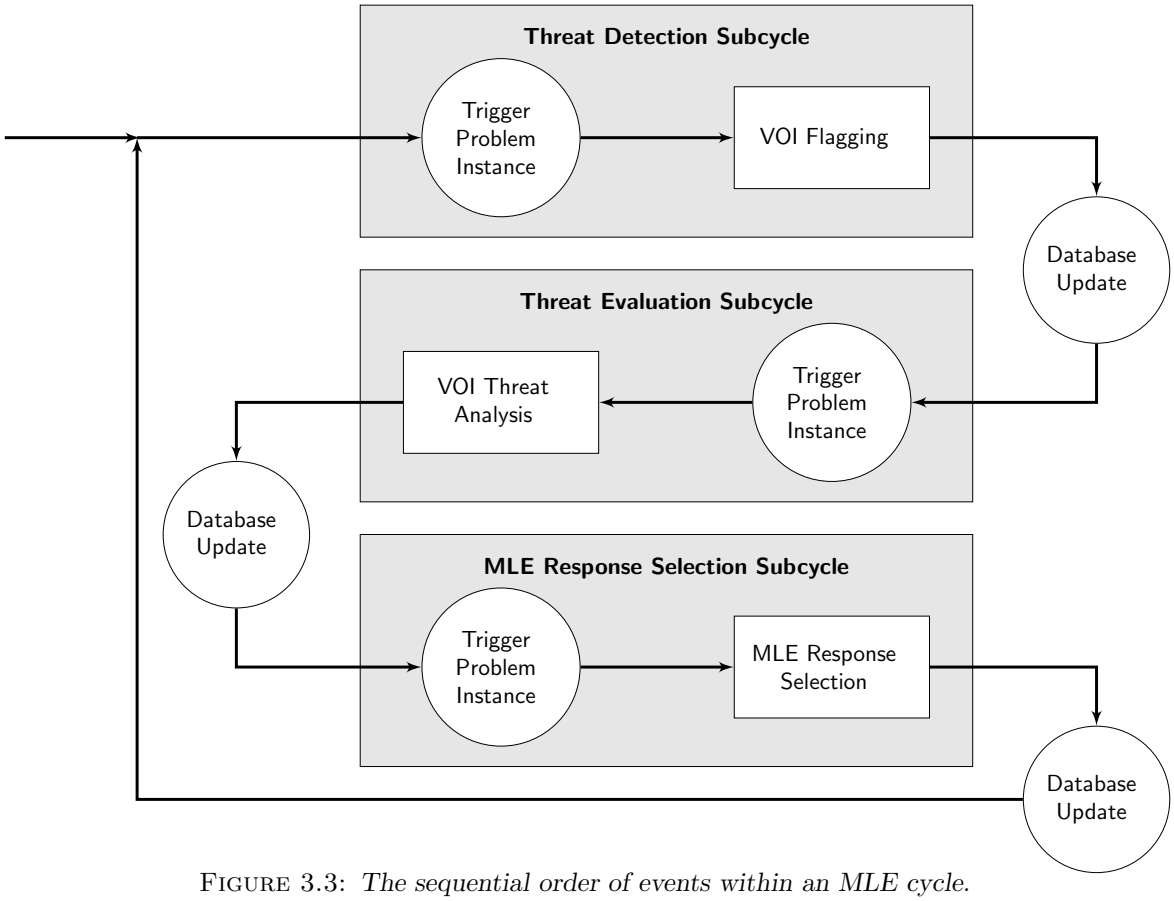


FIGURE 3.3: *The sequential order of events within an MLE cycle.*

Once an MLE threat detection subcycle is triggered, a snapshot of the MLE database is taken at that specific time instant. This snapshot contains the data streams (obtained from the external MLE threat detection infrastructure elements) to be used during the current MLE computation cycle, displayed *via* the HMI. Thereafter, VOI flagging commences during which the objects at sea deemed suspicious are flagged, after which the information associated with the identifiers of these objects are updated in the database after having formatted their contents *via* the MLE database manager.

Upon completion of the threat detection subcycle, the threat evaluation subcycle is triggered, another snapshot of the MLE database is taken and the VOI threat analysis commences, during which VOIs are assessed in terms of the perceived nature and level of threat embodied in their activities. Finally, the updated information associated with the identifiers of the relevant maritime objects are transferred to the database.

Upon completion of the threat evaluation subcycle, the MLE response selection subcycle is finally triggered, another snapshot of the MLE database is taken and the MLE response selection process commences, during which the present situation at sea is assesed and suitable routes and VOI visitation schedules are devised for the available MLE resources, after which the updated information associated with the identifiers of the relevant maritime objects are once again transferred to the database.

3.3 Three MLE response selection paradigms

Ideally, the MLE response selection problem should be solved by a centralised operator, responsible for making all of a coastal nation's MLE response selection related decisions. Alternatively, a group of central operators should have the power to override any MLE response selection-related decisions made by less significant operators, should their decisions be considered in contravention of the fundamental MLE goals and objectives of the coastal nation. As discussed in §3.1.2, however, the MLE operations of coastal nations are typically conducted by a set of semi-independent entities, each possessing their own DSSs, operators and MLE resources. The above-mentioned ideal situation may be approximated closely when these entities not only work together in perfect harmony, but also agree altogether that the purposes of certain decision entities are ultimately more important (with respect to the greater good of the coastal nation) than others. Such an altruistic situation cannot, however, realistically be expected to prevail in practice.

In this section, three paradigms for solving the MLE response selection problem are therefore proposed and described in some detail. These paradigms are:

A *centralised* paradigm, where a single operator dictates how decision entities should assign their MLE resources in terms of routing schedules in real time,

An *intermediate* paradigm, where a central operator considers the preferences of decision entities to a certain extent when assessing routing schedules in real time, and

A *decentralised* paradigm, where a central operator shares the MLE response selection decision making process with operators of the various decision entities in the sense that the central operator represents a neutral party in charge of the VOI assignment to decision entities in real time, and the operators of the various decision entities are, in turn, in charge of establishing routing schedules with respect to the VOIs assigned to them involving their own MLE resources in real time.

It is acknowledged that, while from the collective perspective of an entire coastal nation, the first paradigm above represents an ideal scenario in terms of subsequent solution quality, the latter two paradigms represent more likely and realistic MLE response selection decision paradigms to be found in practice. But it is also acknowledged that the first two paradigms above are expected to suffer in solution quality whenever large volumes of input data have to be processed within short time frames from a central processing unit. If a coastal nation therefore does not possess a central infrastructure with strong processing power, then it may be more beneficial to split the tasks among multiple decentralised processing units, perhaps under the jurisdiction of the various decision making entities.

3.3.1 A centralised decision making paradigm

In the centralised decision making paradigm, it is envisaged that a central operator is responsible for overseeing the entire MLE response selection process of a coastal nation with the assistance of a centralised DSS. The duty of the central operator is to dictate to decision entity operators how to allocate their MLE resources to VOIs in real time. Decision entity operators therefore have no say at any point in the decision making process; their duty is merely to ensure that the operations put forward by the central operator are carried out by their respective decision entities. Decision entity operators are still required, however, to update information in the MLE database in respect of real time developments that are relevant to MLE operations assigned

to them. VOI line-of-sight observation reports or status updates in respect of the functional attributes associated with the MLE resources of an entity are other examples of the duties of decision entity operators in this paradigm. The DSS functional elements within this paradigm, together with the flow of information between them, are illustrated in Figure 3.4. In addition, the sequential order in which the functional elements operate within this MLE response selection paradigm cycle is shown in Figure 3.5.

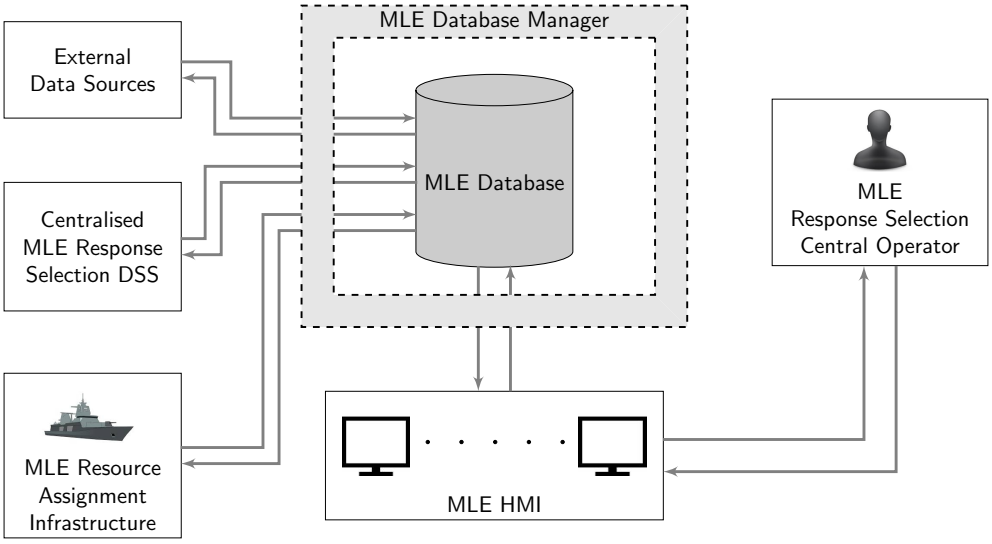


FIGURE 3.4: *Functional elements in the MLE response selection centralised decision making paradigm and the flow of information between these elements.*

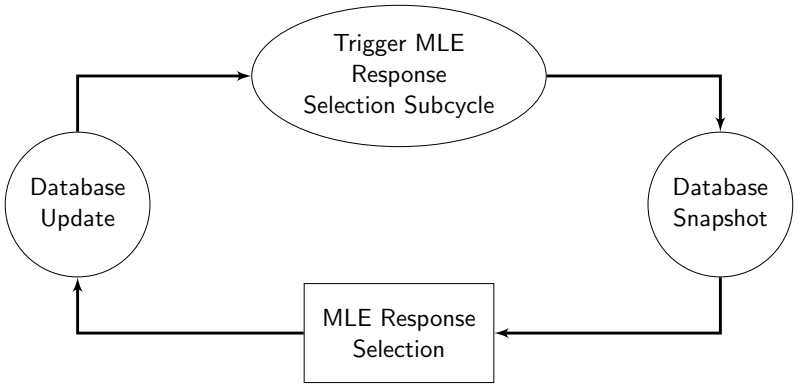


FIGURE 3.5: *The sequential order of events within the MLE response selection centralised decision making paradigm.*

3.3.2 An intermediate decision making paradigm

In an intermediate decision making paradigm, the central operator may choose to consider advice, preferences and expertise from the operators of the various decision entities as to how the MLE response selection routing process should be conducted in real time. The degree of attention that the central operator affords to the opinions of decision entity operators may, however, vary from one scenario to another. Consequently, this paradigm may exist anywhere between the two

extreme paradigms (the centralised paradigm described above and the decentralised paradigm proposed hereafter), possibly under different configurations. The central operator nevertheless has the final authority as to which decision alternative should be implemented.

3.3.3 A decentralised decision making paradigm

In a fully decentralised decision making paradigm, a central operator and multiple decision entity operators are together responsible for overseeing the MLE response selection process of a coastal nation. Decision entities in this paradigm are therefore afforded more decision making power with respect to the MLE response selection process.

In this paradigm it is envisaged that the central operator is responsible, with the assistance of a DSS, to oversee the so-called *VOI distribution* process, which consists of distributing the VOIs amongst the decision entities in real time by associating each decision entity with a specific subset of VOIs that have to be intercepted by it. Additionally, it is assumed that each decision entity has its own operator, routing DSS and MLE resources, and is independently responsible for making routing decisions involving interceptions by its own MLE resources with respect to the subset of VOIs assigned to it. Although decision entities are assumed to operate independently from one another in this paradigm and not share MLE resources, it is assumed that they share coastal nation maritime bases as well as the pre-configured patrol circuits. In this paradigm, the VOI distribution operator is therefore required to solve the VOI distribution problem, after which intercept scheduling and routing decisions are made independently by each decision entity in respect of its own MLE resources for the VOIs assigned to it.

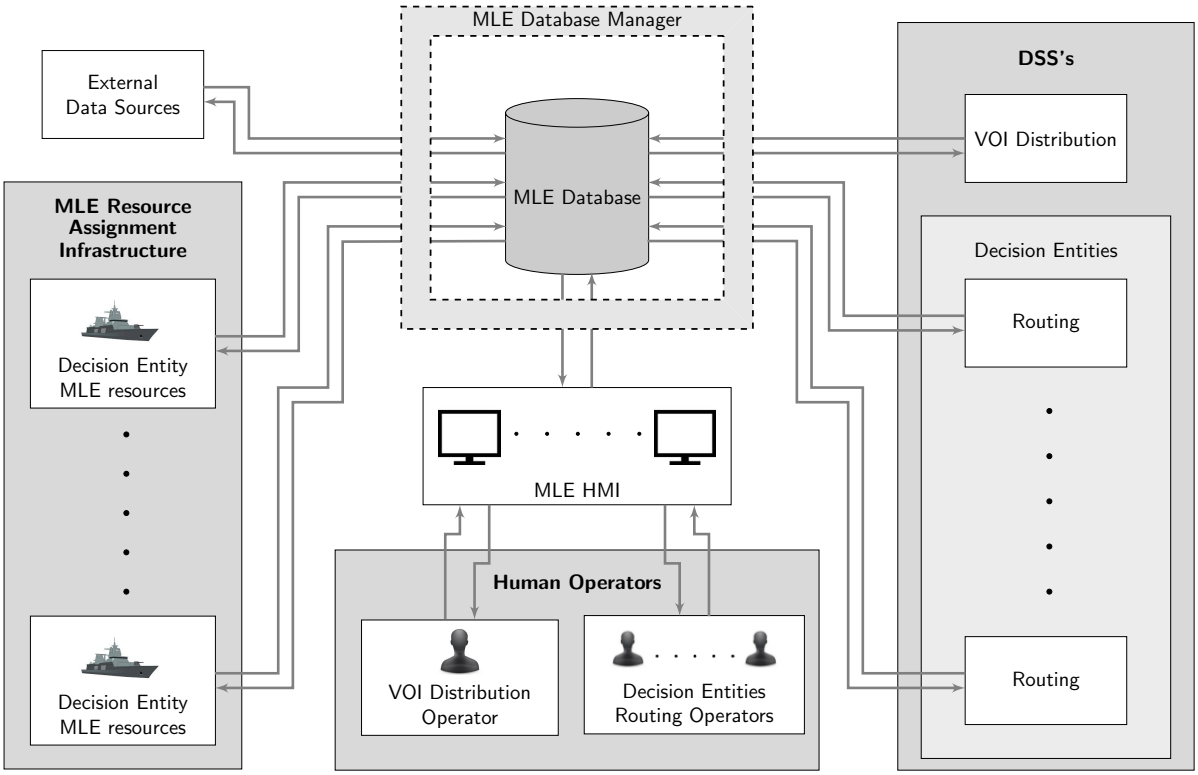


FIGURE 3.6: *Functional elements in the MLE response selection decentralised decision making paradigm and the flow of information between these elements.*

The DSS structure within this paradigm therefore consists of multiple independent operators and DSSs, each responsible for carrying out specific tasks within the MLE response selection process, overseen by a central operator. These operators are, however, assumed to have a local view in respect of their environment, only being concerned with solving their respective parts of the MLE response selection routing problem, and therefore do not need to be aware of the problem specification as a whole. The DSS functional elements within this paradigm, together with the flow of information between them, are illustrated in Figure 3.6. Additionally, the sequential order in which the functional elements function within this MLE response selection paradigm cycle is shown in Figure 3.7.

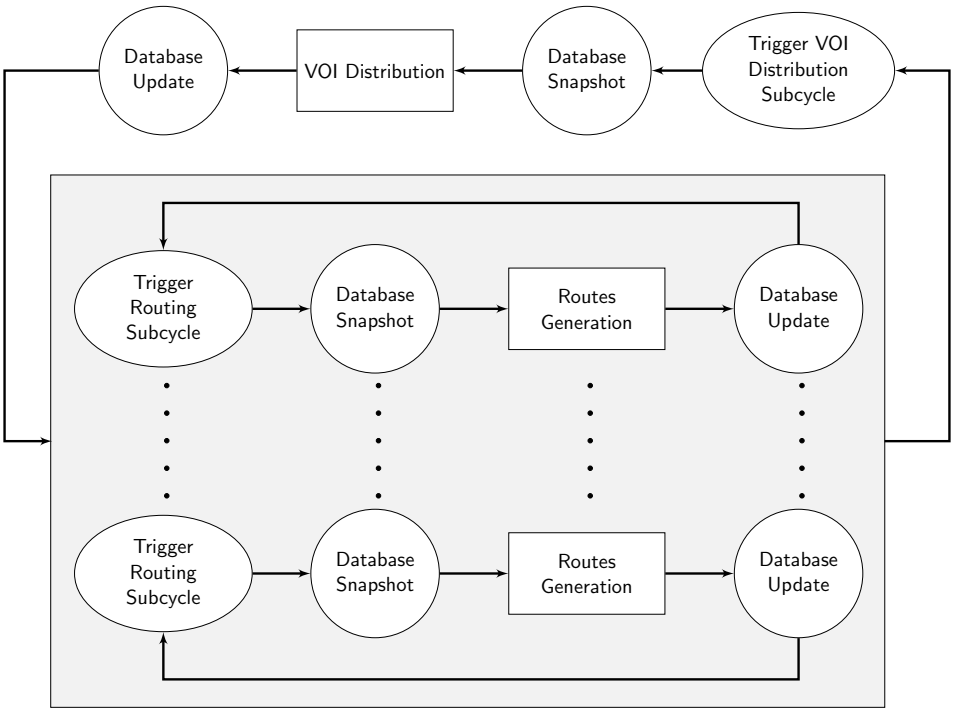


FIGURE 3.7: *The sequential order of events within the MLE response selection decentralised decision making paradigm.*

3.4 Chapter summary

The design of a general system architecture for an MLE DSS was proposed in §3.1 and the functional elements of this architecture were discussed in some detail, with particular emphasis on the MLE response selection external infrastructure functional elements as well as the four subsystems forming the MLE response selection DSS. The flow of events between the threat detection, threat evaluation and MLE response selection subcycles was then highlighted in §3.2. Finally, three generic decision making paradigms were presented in §3.3, capturing the roles of the various decision entities when solving the MLE response selection problem.

The mathematical modelling, model solution and information management subsystem components forming part of the proposed MLE response selection DSS are presented in detail over the course of the next six chapters. First, the model configuration component is presented (in the following chapter). This component is responsible for the construction and storage of fundamental mathematical structures and modelling components within the system.

CHAPTER 4

Model Configuration

Contents

The model selection component of the DSS architecture put forward in Chapter 3 comprises a collection of combinatorial optimisation models, each responsible for solving a special variant of the MLE response selection problem of assigning MLE resources to VOIs in real time. These models are constructed on the basis of fixed and dynamic parameters, variables, fundamental objectives, fundamental constraints and general system dynamism. The selected model structure, referred to as the *fundamental structure*, ultimately affects the nature of the set of alternatives proposed by the DSS as well as the associated complexity of and the time required to solve the model. It is advocated that the operator should not interact with the model selection component on a problem-instance basis, but rather on a long-term basis, as and when certain features (in particular, the values of fixed parameters) of the model may have to be reconsidered.

Various preliminary concepts, ideas and assumptions underlying the MLE response selection problem are presented in §4.1. This is followed in §4.2–4.4 by a description of the modelling

approach adopted in respect of mathematically incorporating each of the three decision making paradigms presented in §3.3 into the DSS, after which the chapter closes with a brief summary in §4.5.

4.1 Concepts and assumptions

Before mathematical models may be developed for incorporation into the DSS framework of Chapter 3, it is necessary to present and define various preliminary concepts and make explicit certain assumptions related to the nature of the MLE response selection problem. First, the notion of reaching a consensus by multiple decision entities is considered in §4.1.1. This is followed by a discussion on the routing characteristics of the MLE response selection problem in §4.1.2, and a discussion on the objectives informing MLE response selection operations in §4.1.3. The expected input data format encapsulating the characteristics of VOIs is then briefly described in §4.1.4, and the notion of the general system dynamism of the MLE response selection problem is formalised in §4.1.5. Discussions on MLE resource counter threat performances, MLE resource autonomy levels, VOI servicing procedures and VOI response times follow in §4.1.6–4.1.9, respectively. Finally, a generic graphical representation of MLE response selection routing operations is presented in §4.1.10.

4.1.1 Reaching a consensus

In the decentralised and intermediate MLE response selection decision making paradigms, newly detected VOIs are individually allocated to decision entities by a central operator. This process can, in essence, be formulated as a multi-person decision making problem, as discussed in §2.1.5, where the proposed alternatives consist of preferential input received from the decision entities and contain information on the desirability levels associated with each VOI in respect of assigning the VOI to a particular decision entity. In other words, decision entities communicate input data with respect to their preferences of being responsible to investigate certain VOIs to the central operator. These preferences are expected to be conflicting in nature and hence a decision that limits the overall level of dissatisfaction amongst the decision entities should ideally be sought (*i.e.* a certain consensus level must be reached). In this context, the notion of reaching a consensus involves the cooperative process of obtaining the maximum degree of agreement between all decision entities and collectively supporting a single solution from a given set of alternatives for the greater good of the coastal nation, as well as achieving certain satisfaction levels of individual decision entities to some (lesser) extent. In addition, the measure of consensus should incorporate the relative importance of the decision entities to the coastal nation, so that preferences expressed by more important decision entities may be prioritised.

In §2.1.5, reference was made to a multi-person decision making approach in which individual preference ordering of alternatives is used, and where each decision maker is required to provide his preferences in respect of a set of alternative solutions (ranked from best to worst) for a given scenario. After the VOI distribution process, however, each decision entity is only concerned with a subset of VOIs, rather than the response selection alternative as a whole (*i.e.* decision entities are assumed to be indifferent towards VOIs assigned to other decision entities). It is therefore suggested that decision entities make known to the central operator their individual VOI distribution preferences in the form of an ordered set of VOIs, ranked from most preferred to least preferred, as well as ideal numbers of VOIs that the various decision entities would like to be assigned.

There are two main reasons for suggesting the specification of decision entity preferences in such a format, rather than requesting ideal subsets of VOIs preferred by each of the decision entities. First, some of the VOIs in such ideal subsets may be more important to the respective decision entity than the other VOIs in those subsets, and the VOI distribution process should account for such relative importance measures. Secondly, it may be the case that VOIs outside the ideal subset expressed by a decision entity are allocated to it, and because this decision entity also has certain preferential levels with respect to those VOIs, the VOI distribution process must also account for this possibility.

4.1.2 Vehicle routing characteristics

In this dissertation, it is proposed that instances of the MLE response selection problem be modelled as a special type of VRP, described in detail in §2.6. In the MLE response selection VRP, the depots represent the bases from whence MLE resources are dispatched, the fleet of vehicles represents the fleet of MLE resources at the collective disposal of the decision entities and the customers represent the VOIs tracked at sea within the territorial waters of the coastal nation. The process of investigating and possibly neutralising VOIs is analogous to *visiting* and *servicing* a customer in the context of a traditional VRP application. Moreover, the notion of patrol circuits in an MLE response selection environment, described in §3.1.2, serves a similar purpose to the *repositioning-and-waiting strategies* in anticipation of future events in the VRP literature [150]. Here, idle vehicles currently situated in low-demand areas are relocated to cover high-demand areas (preferably not too far from a base, depending on the autonomy levels of these vehicles), also known as "areas of interest" or "fruitful regions," based on known *a priori* information in the form of historical data related to previous events.

The MLE response selection problem differs from standard capacitated VRP formulations in the literature as a result of the following characteristics:

1. The fleet of vehicles is not necessarily homogeneous,
2. servicing a customer is optional,
3. assigning a vehicle a route is optional,
4. certain types of vehicles are not suitable to visit certain types of customers,
5. certain types of vehicles are favoured to visit certain types of customers,
6. vehicles are typically not subject to capacity restrictions in the classical sense,
7. the number of customers is not fixed and neither are their locations,
8. the distance between any two customers is not necessarily the same in both directions,
9. the length of a travel arc between any two customers is measured as the shortest possible path linking them aquatically (except in the case of aerial MLE resources),
10. certain situations may require that more than one vehicle be assigned to visit a customer,
11. there may be more than one (inhomogeneous) depot,
12. a vehicle may start out along a route that does not originate or end at a depot,
13. idle vehicles do not have to be stationed at a depot,

14. a vehicle route may start at a certain depot and end at a different one, and

15. certain depots may not necessarily have fixed locations.

In the above list, Characteristics 1–3 follow from the discussion in §2.6 on certain variants of the VRP in the literature. VRPs of the kind mentioned in Characteristic 1 are known in the literature as *VRPs with a Heterogeneous Fleet* [150], while VRPs of the kind mentioned in Characteristic 2 are known in the literature as *VRPs with Customer Profits* [1].

Characteristic 4 places the restriction that certain types of MLE resources should preferably not be scheduled to intercept VOIs that are suspected to be of a certain type, as their infrastructure characteristics may not be effective enough to successfully neutralise threats of that type. Similarly, taking into account the unique properties that are regarded to be advantageous in successfully neutralising threats of certain types, Characteristic 5 encourages certain kinds of MLE resources to preferably intercept VOIs that are suspected to be threats of these types.

Characteristic 6 stresses that the MLE resources are typically not restricted to a capacity constraint (in terms of delivering commodities to customers, as is the case in most VRP problem formulations in the literature), but are rather constrained in terms of maximum distances travelled or lengths of time associated with the routes of the vehicles as a result of various practicalities, such as the need to refuel or resupply materials or ammunition, the need to perform routine MLE resource maintenance and the need to allocate leave or periods of rest to crew members. VRPs of this kind are known in the literature as *Distance-Constrained VRPs* and *Time-Constrained VRPs* [150].

Characteristic 7 acknowledges the kinematic nature of VOIs at sea, calling for interception points to be calculated, while Characteristic 8 allows for the possibility that MLE resource travel directions are influenced by ocean currents and adverse meteorological conditions. VRPs of this kind are known in the literature as *Asymmetric VRPs* [150]. Characteristic 9 refers to the open spherical nature of an oceanic jurisdiction area (barring navigation movement constraints such as land masses, shallows, weather and EEZ boundaries), while Characteristic 10 refers to types of VOIs that do not only involve high threatening intensities, but where it is also deemed necessary to allocate multiple MLE resources to them in order to successfully neutralise them. VRPs of this kind are known in the literature as *VRPs with Split Deliveries*.

Characteristic 11 is applicable to coastal nations having more than one base from whence MLE resources are dispatched. VRPs of this kind are known in the literature as *Multi-Depot VRPs* [150]. In such a case, Characteristic 12 focuses attention on the fact that an MLE resource may be diverted to a new route while out at sea or to a patrol circuit (*i.e.* it is not required that an MLE resource should first return to the base prior to starting out along a new route), and Characteristic 13 allows for the resources to be on stand-by at sea in strategic locations until assigned to a specific interception route. Finally, Characteristic 14 states that certain MLE resources may start out from certain bases and return to other bases, while Characteristic 15 refers to coastal nations possessing certain types of mobile bases which may be deployed at sea to provide basic services to certain subsets of MLE resources.

4.1.3 The objectives informing MLE response selection decisions

As hinted previously, it is proposed that the MLE response selection problem should be modelled as a *multiobjective* VRP. Coastal nations (and their various decision entities) typically have their own values, preferences and perceptions of the desirability of trade-offs between objectives when dealing with threats at sea. It is therefore expected that the MLE response selections of the

various role players will vary from one to another. These responses should, however, be coherent and be carried out according to a pre-determined protocol, based on a set of goals and objectives, as discussed in §2.1. Identifying and understanding these fundamental objectives is critical in the design of an MLE response selection decision support system, as they provide a means of measuring the quality of solutions as perceived by the coastal nation and its decision entities. These objectives should therefore be derived from the subjective preferences of the specific coastal nation together with those of its decision entities.

For security reasons, coastal nations and their decision entities are typically averse to declaring their MLE response policies and strategies in detail in the open literature. For this reason, the current discussion leans to a generic approach towards identifying plausible and realistic objectives in pursuit of suitable solutions to MLE response selection problem instances. It is, however, acknowledged that a deep understanding of the specific strategic aims of a coastal nation's MLE efforts (as well as those of its individual decision entities) is necessary in order to identify a suitable set of fundamental objectives for use in the formulation of MLE response selection strategy. Moreover, it should be noted that the model objectives proposed in this dissertation do not all have to be incorporated in the decision making process; the decision maker may select a subset of (or adopt other) objectives for consideration in the decision making process.

4.1.4 VOI input data

The nature and frequency of maritime threats detected vary in different regions of the world, and different coastal nations typically face different types of threats at different levels of harm or intensity. The degree of effectiveness of the threat detection and threat evaluation subsystems described in the previous chapter plays a critical role in providing accurate and complete input data facilitating the efficient functioning of an MLE response selection DSS.

Two important assumptions are made in this chapter with respect to these input data. First, it is assumed that the position and velocity of each VOI is known at all times (these values may, of course, change over time in a stochastic manner), and that estimated itineraries are available for all VOIs. Secondly, the threat evaluation operator cannot always know with certainty the potential threat that is associated with a VOI under investigation. Therefore, a distribution of approximate probabilities matching each VOI with each type of threat, a type of unknown threat and a false alarm (*i.e.* the probability that a VOI does not embody any threat), is assumed to form part of the problem input data to the MLE response selection problem, as provided by the threat evaluation subsystem.

4.1.5 Disturbances and time stages

Once an acceptable solution to the MLE response selection problem has been found, it is merely a matter of time before the situation at sea requires reconsideration. Input data to problem instances are made known to the operator/decision maker in a continual fashion and are updated concurrently with the determination of a problem solution. This concept is better known as *general system dynamism*, as described in §2.6.3. In other words, the input data of the MLE response selection DSS are not known in their entirety ahead of time, but are rather revealed as time goes by. An operator must therefore solve part of the problem on the basis of the information currently available, and then re-solve part of the problem as new input data are revealed, since parts of the problem solution might no longer be feasible or preferred. It follows that the MLE response selection routing problem may be classified as a dynamic VRP.

In order to accommodate the dynamic nature of the MLE response selection problem, a *disturbance* is defined as a threshold phenomenon occurring stochastically in time which may cause the currently implemented solution to suffer in terms of quality and/or feasibility, hence triggering the start of a new problem instance during which the current situation is re-evaluated (*i.e.* the instance is re-solved under the original information combined with the data update which brought along the disturbance, disregarding those VOIs of the previous solution that have already been intercepted). Henceforth, define the notion of a *time stage* as a finite temporal interval that marks the start and finish of a problem instance. A disturbance therefore causes the current time stage to end and the next one to begin.

4.1.6 MLE resources counter-threat performances

As a result of their unique infrastructure, size, crew expertise and speed, certain types of MLE resources excel at countering VOIs embodying certain types of threats. This is particularly the case if a VOI embodies a type of threat for which the decision entity in charge of the MLE resource scheduled to intercept this VOI specialises in countering threats of that type. Ensuring that the right MLE resources are dispatched to intercept VOIs is beneficial for many reasons. For instance, the ability to counter certain threats effectively keeps the danger of retaliations, material damage and possible human injuries to a minimum. Furthermore, effectively investigating and/or neutralising VOIs reduces the service time. This is a crucial factor to accommodate whenever the MLE resource has one or more VOIs to investigate on its visitation route after having intercepted the current customer[1].

It may be the case that a certain type of MLE resource does not only have a low efficiency value in respect of countering certain types of threats, but also does not have the capacity to successfully neutralise VOIs embodying certain types of threats. An MLE resource is defined to be *strictly incapable* of neutralising a certain VOI if it is forbidden to attempt to neutralise a VOI or incapable of neutralising a VOI embodying a (known) specific type of threat. Situations in which MLE resources are assigned to VOIs that they are strictly incapable of neutralising are defined as *infeasible encounters*[2]. Infeasible encounters would never occur if the threat associated with any VOI were known in advance (assuming that the MLE response selection routing operator is competent). As discussed in §4.1.4, however, expectations with respect to the nature of VOIs are subject to error and, although the nature of certain VOIs are easily detected and evaluated in respect of the threat that they embody, others may carry a high level of uncertainty, in turn increasing the risk of infeasible encounters. The degradation in MLE response selection operations caused by infeasible encounters may be measured by a penalty function.

4.1.7 MLE resources autonomy levels

As stated in Characteristic 6 of §4.1.2, it is suggested that the MLE response selection problem is, *inter alia*, formulated as a distance and time constrained VRP. Define the autonomy level of an MLE resource with respect to distance, called its *distance autonomy level*, as the maximum distance that it may travel at sea before having to return to a designated base. Similarly, define the autonomy level of an MLE resource with respect to time, called its *time autonomy level*, as the maximum time that it may spend at sea before having to return to a designated base.

[1] The VOIs tracked at any time should generally be dealt with as quickly as possible to lower the chances of their escape from the territorial waters of the coastal nation or losing their tracks on the radar.

[2] Infeasible encounters are analogous to so-called *service failures* in the VRP literature [150].

It is assumed that, while the distance autonomy level of an MLE resource only diminishes while it is in motion at sea (either in an active or an idle state), its time autonomy level diminishes continually over time from the moment the MLE resource leaves a base. It should therefore be noted that, if an MLE resource starts out along a route that does not originate at a depot (see Characteristic 12 of §4.1.2), its initial autonomy levels will be lower than they would normally be had the same MLE resource departed from a base. Once at a base, MLE resources are replenished over some period of time and their autonomy levels are reset to their maximum levels.

Due to possible distance-constrained feasibility concerns, and because the idle MLE resources management operator cannot know *a priori* where active MLE resources will be located in space after investigating the last VOIs on their routes, so-called *patrol autonomy thresholds* are incorporated into the model formulation proposed in this dissertation. These thresholds ensure that an MLE resource is only allowed to join a patrol circuit after having completed its mission, provided that the travel distance and time to the circuit are within specific autonomy levels. Ultimately, a route may only be classified as distance-constrained feasible or time-constrained feasible if there exists at least one approved base that is at most as far away (in both distance and time) as the calculated autonomy level threshold associated with the MLE resource after having investigated the last VOI on its route. Lastly, *route failure* is said to occur whenever an MLE resource is unable to complete a route due to unpredictable developments increasing the route autonomy.

4.1.8 VOI servicing

As discussed in §4.1.2, MLE response selection vehicle routing comprises the scheduling of MLE resources to visitation routes containing one or more VOIs, from where the visited VOIs are subjected to an investigation and possible neutralisation if they embody threats. Moreover, the *service time* of a VOI may be described as the time elapsed between the moment an MLE resource intercepts it at sea to the moment it completes servicing it.

The service time of a VOI cannot, however, be known *a priori*, as characteristics associated with VOIs (even those embodying the same threatening activities) may significantly vary from one VOI to another. In addition, and as discussed in §4.1.6, certain MLE resources are better suited to neutralise VOIs embodying certain types of threats, which is an important factor to take into account when assessing service time. As a result, the service times of VOIs have to be estimated based on the distributions of approximate threat probabilities associated with each VOI (as described in §4.1.4) and historical data pertaining to the expected service times of certain MLE resources in respect of VOIs embodying certain threats.

Because disturbances occur stochastically in time, it may be the case that a new time stage is triggered while certain active MLE resources are in the process of servicing VOIs. Consequently, it is assumed that VOIs which are in the process of being serviced at the end of a time stage, along with their estimated remaining service times, are carried over to the next time stage.

4.1.9 VOI response time

The *response time* of a VOI within a certain time stage is defined as the time elapsed between the detection of the VOI and its estimated interception time, provided that it is scheduled to be intercepted during that time stage (see Characteristic 2 of §4.1.2). The response time of a VOI may therefore be estimated by adding together (1) the time elapsed since it was first detected

prior to the current time stage, (2) the estimated travel time along the visitation route of the VOI during the current time stage prior to its interception, (3) the estimated service times of preceding VOIs visited along the same visitation route of the VOI during the current time stage and (4) the estimated set-up time of the MLE resource scheduled to intercept the VOI, provided that it was idle at a base at the end of the previous time stage.

4.1.10 A graphical representation of MLE response selection

For any given time stage, the vertices in an MLE response selection environment may be partitioned into four sets: VOIs, MLE resources (both active and idle ones), patrol circuits and bases. While the set of VOIs is typically updated at the start of every time stage due to its high level of dynamism, the other three sets remain somewhat more fixed and are updated independently from time stages (with occasional exceptions, such as a disturbance caused by an active MLE resource breaking down at sea).

Henceforth, let $\mathcal{V}_\tau^e = \{1, \ldots, n_\tau\}$ represent the set of VOIs at the beginning of time stage τ, let $\mathcal{V}^r = \{1, \ldots, m\}$ be the set of available MLE resources[3], let $\mathcal{V}^b = \{1, \ldots, |\mathcal{V}^b|\}$ denote the set of bases and let $\mathcal{V}^p = \{1, \ldots, |\mathcal{V}^p|\}$ represent a set of pre-determined patrol circuits. Additionally, let $\mathcal{V}_\tau = \mathcal{V}_\tau^e \cup \mathcal{V}^r \cup \mathcal{V}^b \cup \mathcal{V}^p$. These vertex subsets, along with the arcs inter-linking them, form the directed graph $G(\mathcal{V}_\tau, \mathcal{E}_\tau)$ depicted in Figure 4.1, where $\mathcal{E}_\tau$ is the set of pre-calculated arcs linking the vertices in such a way that all pairs of vertices in $\mathcal{V}_\tau^e$ are reachable from one another. Finally, let $\mathcal{V}_{k\tau}^e \subseteq \mathcal{V}_\tau^e$ be the ordered set of VOIs scheduled to be investigated by MLE resource k during time stage τ. In this model formulation, active MLE resources in $\mathcal{V}^r$ are assigned to investigate subsets of VOIs in $\mathcal{V}_\tau^e$ during time stage τ, after which they are assigned to either travel back to a base in $\mathcal{V}^b$ or to a designated patrol circuit in $\mathcal{V}^p$.

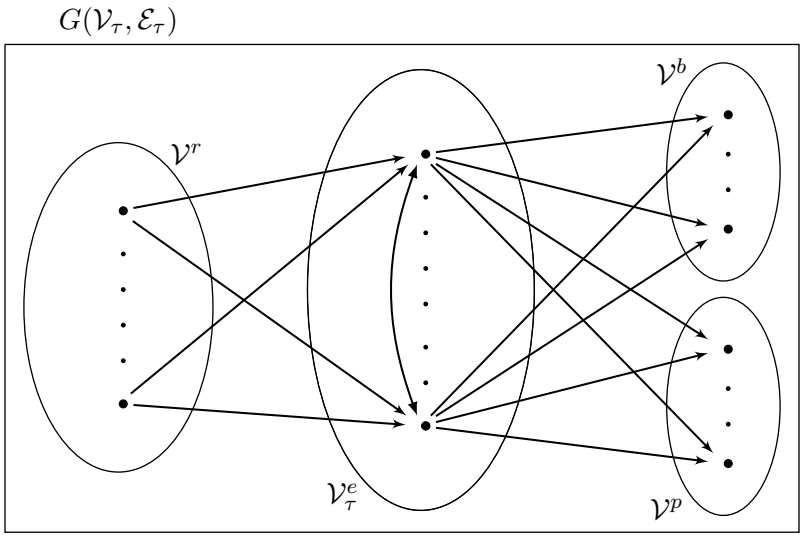


FIGURE 4.1: *Graph representation of routing decisions in an MLE response selection environment.*

[3]The set of available MLE resources is generally fixed in the long run and is usually not affected by the more common types of disturbances.

4.2 Model configuration in the centralised paradigm

In this section, a number of fundamental elements are put forward in respect of modelling the MLE response selection problem in the centralised decision making paradigm (see §3.3.1). Due to the complex nature of the problem, the constituent elements of the model formulation are partitioned into six groups, namely parameters (both fixed and dynamic), variables, objectives, fundamental constraints, other constraints and system dynamism.

4.2.1 Model parameters

Following the discussion on VOI input data in §4.1.4, the detection system of a coastal nation tracks, during time stage τ, n_τ VOIs, which are individually matched with estimated probabilities to each of $|\mathcal{H}|-2$ known threat classes, an unknown threat class and a false alarm class. If $p_{ih\tau}$ is the probability at time τ that VOI i falls in class h, then, of course,

$$\sum_{h=1}^{|\mathcal{H}|} p_{ih\tau} = 1, \qquad i \in \{1, 2, \ldots, n_\tau\}.$$

It is assumed that the position and velocity of each VOI is known at all times (these values may, of course, change over time in a stochastic manner), and that estimated itineraries are available for all VOIs.

Let $d_{ijk\tau}$ be the length of the arc $(i,j)_\tau$ in the graph of Figure 4.1 traversed by MLE resource k during time stage τ and set $d_{iik\tau} = +\infty$ for all vertices in $\mathcal{V}_\tau$. If it is assumed that an MLE resource k maintains a fixed average speed of η_k between any two vertices in $\mathcal{V}_\tau$, then the time that MLE resource k spends traversing the arc $(i,j)_\tau$ may be expressed as $\frac{d_{ijk\tau}}{\eta_k}$.

Following the discussion in §4.1.7, let $a^d_{k\tau}$ be the distance autonomy level of MLE resource k at the start of time stage τ, and let $a^t_{k\tau}$ be its time autonomy level at the beginning of time stage τ. In addition, define the fixed parameters $A^d_{k\rho}$ and $A^t_{k\rho}$ as the distance and time patrol autonomy thresholds (respectively) that MLE resource k is at least required to have after visiting the last VOI on its route in order to be allowed to be allocated to travel to patrol circuit $\rho \in \mathcal{V}^p$ at the end of its route during time stage τ. If a certain MLE resource may never be allocated to a certain patrol circuit during any time stage $\tau \in \mathbb{N}$, then the corresponding patrol autonomy threshold value is set to $+\infty$.

Following the discussion in §4.1.8, define the parameter S^t_{kh} as the expected service time that MLE resource k will take to investigate and/or neutralise VOIs embodying a threat type h. Then

$$S_{ik\tau} = \sum_{h \in H} S^t_{kh} p_{ih\tau}$$

is the expected *service time* of VOI i, provided that it is scheduled to be visited by MLE resource k during time stage τ. Furthermore, as has already been mentioned, there may be cases where a VOI is already in the process of being serviced when a time stage ends. Thus, given that VOI i is in the process of being serviced when a new time stage is triggered, let $S'_{i\tau}$ be the expected remaining time required to complete this service[4] and, in this case, set $S_{ik\tau} = S'_{i\tau}$.

Following the discussion in §4.1.9, Let $T^d_{i\tau}$ be the time elapsed between the detection of VOI i and the start of time stage τ, let T^w_k be the expected set-up time incurred when preparing MLE

[4]This estimated service time is relatively easily calculated, as the type of threat embodied by the VOI is completely revealed at that point in time.

resource k for a mission, and define the parameter

$$\gamma_{k\tau} = \begin{cases} 1, & \text{if MLE resource } k \text{ is not on stand-by at the end of time stage } \tau - 1 \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, let $T^t_{ik\tau}$ denote the expected transit times accumulated by MLE resource k along its visitation route from the start of time stage τ until it reaches VOI i. Similarly, let $T^s_{ik\tau}$ denote the expected VOI service times accumulated by MLE resource k along its visitation route from the start of time stage τ until it services VOI i. Then

$$t_{i\tau} = T^d_{i\tau} + T^t_{ik\tau} + T^s_{ik\tau} + \gamma_{k\tau} T^w_k$$

is the expected response time of VOI i during time stage τ, where

$$T^t_{ik\tau} = \begin{cases} \frac{d_{0_{k\tau} i k\tau}}{\eta_k}, & \text{if } i = \mathcal{V}^e_{k\tau}(1) \\ \frac{d_{0_{k\tau}\mathcal{V}^e_{k\tau}(1)k\tau} + \sum_{j=2}^{|\mathcal{V}^e_{k\tau}(i)|} d_{\mathcal{V}^e_{k\tau}(j-1)\mathcal{V}^e_{k\tau}(j)k\tau}}{\eta_k}, & \text{otherwise,} \end{cases}$$

and

$$T^s_{ik\tau} = \begin{cases} 0, & \text{if } i = \mathcal{V}^e_{k\tau}(1) \\ \sum_{\ell=2}^{|\mathcal{V}^e_{k\tau}(i)|} S_{\mathcal{V}^e_{k\tau}(\ell-1)k\tau}, & \text{otherwise.} \end{cases}$$

Here, the notation $\mathcal{V}^e_{k\tau}(i)$ represents the i^{th} VOI on the route of MLE resource k during time stage τ while $|\mathcal{V}^e_{k\tau}(i)|$ is a cardinal number representing the ordered position of that VOI on the route. For example, if $\mathcal{V}^e_{k\tau} = \{5, 16, 9\}$, then $\mathcal{V}^e_{k\tau}(3) = 9$ and $|\mathcal{V}^e_{k\tau}(9)| = 3$. Furthermore, $0_{k\tau}$ represents the initial position of MLE resource k at the beginning of the time stage. Finally, set $t_{i\tau} = 0$ if VOI i was in the process of being serviced by a MLE resource when time stage $\tau - 1$ ended. Of course, $t_{i\tau}$ is set to 0 if it is not scheduled as part of any visitation route.

Next, the monetary cost associated with MLE resource k traversing the arc $(i, j)_\tau$ in the graph of Figure 4.1 during time stage τ is assumed to be estimated as $\Gamma_k d_{ijk\tau}$, where Γ_k is the cost per unit distance travelled associated with MLE resource k. Note that $\Gamma_k d_{iik\tau} = +\infty$ holds for all vertices in $\mathcal{V}_\tau$. In addition, define the set-up cost C^s_k incurred when preparing MLE resource k for departure on a mission. Moreover, costs may be associated with trajectory deviations from one route to another due to factors such as operational manoeuvres or disruption of the original solution due to a disturbance. Hence, define the disruption cost C^d_k to be incurred if MLE resource k was active at the end of time stage $\tau - 1$ and was subsequently scheduled to a new trajectory (*i.e.* sent on a different route) at the beginning of time stage τ (noting that such a cost should not be fixed, but rather expressed as a function of the intensity of the disruption for that particular situation). In relation to this last cost, let $J_{k(\tau-1)} \in \mathcal{V}^e_\tau$ be the next VOI that was scheduled to be intercepted by MLE resource k at the end of time stage $\tau - 1$ (provided that the MLE resource was active then), or set $J_{k(\tau-1)}$ equal to the dummy value 0 otherwise (noting that $0 \notin \mathcal{V}^e_\tau$ for all $\tau \in \mathbb{N}$).

Following the discussion in §4.1.6 and Characteristics 4 and 5 of §4.1.2, it is furthermore necessary to establish parametric values for the counter-threat performance of MLE resources. In this respect, let $W_{kh} \in [0, 1]$ be the score associated with the efficiency of MLE resource k in terms of neutralising a class h threat. In addition, the nature and frequency of maritime threats detected vary in different regions of the world, and different coastal nations typically face different types of threats at different levels of harm or intensity. Thus, let the score $Q_h \in [0, 1]$ associated with class h threats represent the priority level assigned by the coastal nation to neutralise threats

in this class. Lastly, define the fixed parameter

$$\delta_{kh} = \begin{cases} 1, & \text{if MLE resource } k \text{ is strictly incapable of neutralising VOIs of threat type } h, \\ 0, & \text{otherwise} \end{cases}$$

and let $C^e_{ikh\tau}$ be the estimated cost of a potentially infeasible encounter between VOI i and MLE resource k during time stage τ, if this VOI were to embody a type threat h. Of course, $C^e_{ikh\tau} = 0$ whenever $\delta_{kh} = 0$, and $W_{kh} = 0$ whenever $\delta_{kh} = 1$.

Finally, as discussed in §3.1.2, MLE bases typically differ from one another in respect of numerous physical characteristics that are relevant to the MLE response selection problem, such as infrastructure size, facilities available, accessibility and spatial location. These characteristics may impose certain end-of-route restrictions on the allocation of MLE resources to bases, preventing certain MLE resources from being scheduled at any time to end their routes at certain bases. To prevent infeasible assignments of this kind, the parameter

$$\beta_{bk} = \begin{cases} 1, & \text{if MLE resource } k \text{ is allowed to be scheduled to end its route at base } b \\ 0, & \text{otherwise} \end{cases}$$

is introduced, where $k \in \mathcal{V}^r$ and $b \in \mathcal{V}^b$.

4.2.2 Decision variables

In the model formulation to follow, the decision variables

$$x_{ijk\tau} = \begin{cases} 1, & \text{if MLE resource } k \text{ is scheduled to traverse the arc } (i,j)_\tau \text{ in Figure 4.1 ,} \\ 0, & \text{otherwise} \end{cases}$$

and

$$y_{ik\tau} = \begin{cases} 1, & \text{if MLE resource } k \text{ is scheduled to visit VOI } i \text{ during time stage } \tau \text{ ,} \\ 0, & \text{otherwise} \end{cases}$$

are applicable.

4.2.3 Model objectives

A selection from the following set of objectives is proposed to inform the formulation of MLE response strategies in the centralised paradigm:

I Maximise the combined *visitation score* of VOIs scheduled to be intercepted and investigated, weighted by (a) the probabilities of these VOIs being various types of threats from a pre-specified list of relevant threats and (b) the priorities assigned by the coastal nation with respect to neutralising the threat types from this list.

II Minimise the combined *time delay score* of VOIs scheduled to be intercepted, weighted in the same manner as in Objective I above.

III Minimise the total *operating costs* incurred by the dispatch of active MLE resources, including (a) their set-up costs, (b) their travelling costs while in an active state and (c) their trajectory deviation costs.

IV Minimise the so-called *incapacity score* of MLE resources obtained from the accumulated risk of infeasible encounters between active MLE resources and the VOIs scheduled for interception on their routes, weighted in the same manner as in Objective I above as well as with an estimated opportunity cost resulting from the inability of the MLE resources to neutralise these VOIs.

Mathematically, these objectives may respectively be formulated as follows:

I
$$\text{Maximise} \quad \sum_{i\in\mathcal{V}_\tau^e}\sum_{k\in\mathcal{V}^r} y_{ik\tau}\sum_{h\in\mathcal{H}} Q_h W_{kh} p_{ih\tau},$$

II
$$\text{Minimise} \quad \sum_{i\in\mathcal{V}_\tau^e} t_{i\tau}\sum_{k\in\mathcal{V}^r} y_{ik\tau}\sum_{h\in\mathcal{H}} p_{ih\tau} Q_h,$$

III
$$\text{Minimise} \quad \sum_{k\in\mathcal{V}^r}\left(\Upsilon^s_{k\tau}+\Upsilon^t_{k\tau}+\Upsilon^p_{k\tau}\right), \text{ and}$$

IV
$$\text{Minimise} \quad \sum_{i\in\mathcal{V}_\tau^e}\sum_{k\in\mathcal{V}^r} y_{ik\tau}\sum_{h\in\mathcal{H}} \delta_{kh} p_{ih\tau} C^e_{ikh\tau},$$

where in Objective III

$$\Upsilon^s_{k\tau} = \gamma_{k\tau} C^s_k \sum_{j\in\mathcal{V}_\tau^e} x_{0_{k\tau}jk\tau},$$

$$\Upsilon^t_{k\tau} = \Gamma_k \sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}}\sum_{\substack{j\in\mathcal{V}_\tau\\ j\neq 0_{k\tau}}} x_{ijk\tau} d_{ijk\tau}$$

and

$$\Upsilon^p_{k\tau} = C^d_k(1 - x_{0_{k\tau}J_{k(\tau-1)}k\tau}) \sum_{\substack{j\in\mathcal{V}_\tau^e\\ j\neq J_{k(\tau-1)}}} x_{0_{k\tau}jk\tau}.$$

Objective I refers to the significance of the expected benefits associated with the successful interception and neutralisation of various types of threats at sea, as discussed in §4.1.6. It is assumed that the level of threat posed by a VOI at sea is directly proportional to both the probability that the VOI is actually a threat of a specific type and the importance of neutralising such a type of threat. Pursuit of this objective is expected to increase the probability of assigning suitable or highly appropriate MLE resources to intercept VOIs that are expected to embody highly dangerous threats. This objective is expressed as a linear combination of the probabilities of VOIs being the types of threats mentioned in §4.1.4 and the priority scores associated with neutralising threats of these types.

The second objective above refers to the total time that elapses between the detection and interception of the VOIs at sea that are scheduled for interception. It is assumed that taking long to intercept a VOI (*i.e.* incurring a long MLE response time) impacts negatively on the overall success of MLE operations. More specifically, a longer response time is expected to increase the chance that tracking operators of the VOI fail or that the VOI exits the EEZ before interception takes place. This objective is expressed as a linear combination of the probabilities of VOIs being the types of threats mentioned in §4.1.4, the priority scores associated with neutralising threats

of these types and the expected response times of the MLE resources assigned to investigate the VOIs. Pursuit of this objective is therefore expected to result in seemingly more threatening VOIs being serviced more rapidly.

Objective III refers to the costliness of committing MLE resources to investigating VOIs at sea. As mentioned previously, it is assumed that an MLE resource incurs a fixed set-up cost every time it is dispatched from a base and a variable operating cost which depends on the overall distance travelled on a mission at sea. Furthermore, it is assumed that an active MLE resource incurs a certain cost associated with the manoeuvres required when changing routes at the start of a problem instance. In the mathematical formulation of this objective,

Objective IV above relates to the possibility of infeasible encounters between MLE resources and VOIs. Such encounters should, of course, be avoided in MLE resource assignment operations, if possible, as they result in wasted travel costs and unnecessary VOI servicing delays. The objective is expressed as a linear combination of the probabilities of VOIs embodying certain types of threats and the estimated costs associated with infeasible assignments.

Objectives I and II share a certain degree of inter-dependency. It should, however, be observed that these two objectives complement each other in a manner that would not be possible if they were to be pursued in isolation. On the one hand, solely adopting Objective I may lead to increased levels of interception of those VOIs expected to embody highly dangerous threats, but will not account for the urgency factor involved when planning the trips of the MLE resources assigned to carry out these interceptions. On the other hand, solely adopting Objective II will lead to the situation where no VOIs are actually intercepted (the situation of a zero delay score) and would analogously benefit Objective III as well.

Additionally, Objectives I and IV share a certain degree of inter-dependency, as the visitation score is expected to discourage the assignment of MLE resources for the interception of VOIs embodying types of threats which they are strictly incapable of neutralising. Because the nature of a VOI is, however, typically not known in advance with certainty, there may be a (typically relatively small) probability that the VOI embodies a certain type of threat which cannot be neutralised by certain MLE resources. Because the visitation score will afford more weight to the scores with higher probabilities, these small probability terms may, in fact, not have a significant impact on the overall visitation score, thus prompting the suggestion of Objective IV above. There are two main scenarios in which this objective is anticipated to be especially relevant. It may be the case that the costs associated with infeasible encounters are high (*e.g.* as a result of a large jurisdiction area, limitations in terms of the number and/or strength of MLE resources, the presence of numerous types of threats to deal with, or poor threat evaluation performances). In such a case Objective IV will promote a minimisation of the probability of infeasible encounters which may not be avoided when only implementing Objective I (which does not directly account for the lost opportunity as a result of infeasible encounters). The operator may indeed be interested in selecting solutions which embody low risks of producing infeasible encounters, while also achieving good visitation scores. Secondly, although the operator may choose to implement Objective I in the model, he may lean more towards implementing solutions which incur low operating costs (see Objective III).

Because Objective IV also deals with costs to some extent, these two objectives may be perceived as correlated. In other words, if the operator has strong preferences for implementing solutions with low operating costs, then he will also prefer to implement solutions with low risks of infeasible encounters, so as to minimise the overall economical implications of both objectives.

4.2.4 Fundamental model constraints

The foremost constraint in this model involves the coupling of the routing variables as

$$\sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}}\sum_{\substack{j\in\mathcal{V}_\tau\\ j\neq 0_{k\tau}}} x_{ijk\tau} \;=\; \sum_{\ell\in\mathcal{V}_\tau^e} y_{\ell k\tau}, \qquad k\in\mathcal{V}^r,$$

which ensures that MLE resource k is scheduled to a visitation route containing VOI i if and only if there exists an arc entering this VOI position which is traversed by the MLE resource. Moreover, the constraint set

$$\sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}} x_{ijk\tau} - \sum_{\substack{\ell\in\mathcal{V}_\tau\\ \ell\neq 0_{k\tau}}} x_{j\ell k\tau} \;=\; 0, \qquad j\in\mathcal{V}_\tau^e,\; k\in\mathcal{V}^r$$

ensures that, if a VOI is intercepted by a certain MLE resource during time stage τ, then that MLE resource must depart from the same VOI after investigation. Lastly, the occurrence of subtours in the visitation process of VOIs is prohibited by means of the constraint set

$$\sum_{i\in\mathcal{V}_\tau^e}\sum_{\substack{j\in\mathcal{V}_\tau^e\\ j\neq i}} x_{ijk\tau} \;\leq\; |\mathcal{V}_{k\tau}^e| - 1, \qquad |\mathcal{V}_{k\tau}^e|\geq 2,\; k\in\mathcal{V}^r.$$

Because the VRP formulation of the MLE response selection problem proposed in this dissertation includes customer profits, not all VOIs have to be included in visitation routes (see Characteristic 2 of §4.1.2). Moreover, it is assumed that no more than one MLE resource may be assigned to investigate the same VOI. Consequently, the constraint set

$$\sum_{k\in\mathcal{V}^r} y_{ik\tau} \;\leq\; 1, \qquad i\in\mathcal{V}_\tau^e$$

ensures that no more than one arc may enter or leave any particular VOI along an MLE resource route in the graph of Figure 4.1.

The framework of §4.1.10 assumes that all active MLE resources are scheduled to end their routes at either a base or a patrol circuit. In other words, the number of MLE resources assigned to routes during time stage τ (*i.e.* the number of active MLE resources) coincides with the number of arcs entering both the base set and the patrol circuit set in the graph of Figure 4.1. One way to implement this condition is to include the constraint

$$\sum_{j\in\mathcal{V}_\tau^e}\sum_{k\in\mathcal{V}^r} x_{0_{k\tau}jk\tau} \;=\; \sum_{i\in\mathcal{V}_\tau^e}\sum_{\ell\in\mathcal{V}^b\cap\mathcal{V}^p}\sum_{k\in\mathcal{V}^r} x_{i\ell k\tau}$$

in the model formulation.

Next, following the discussion in §4.1.2, MLE resources are both distance and time constrained, as they may only travel a certain period of time and cover a maximum distance before they have to relocate to a base. The constraint set

$$\sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}}\sum_{\substack{j\in\mathcal{V}_\tau\\ j\neq 0_{k\tau}}} d_{ijk\tau}x_{ijk\tau} \;\leq\; a_{k\tau}^d, \qquad k\in\mathcal{V}^r$$

therefore imposes an upper bound on the maximum distance travelled by MLE resource k from the beginning of time stage τ, while the constraint set

$$\sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}}\sum_{\substack{j\in\mathcal{V}_\tau\\ j\neq 0_{k\tau}}} \frac{d_{ijk\tau}}{\eta_k}x_{ijk\tau} \;\leq\; a_{k\tau}^t, \qquad k\in\mathcal{V}^r$$

imposes a similar upper bound on the maximum length of time that MLE resource k spends at sea from the beginning of time stage τ.

With regard to post-mission assignments, the set of constraints

$$\sum_{i \in \mathcal{V}_\tau^e} x_{ibk\tau} \leq \beta_{bk}, \quad b \in \mathcal{V}^b,\ k \in \mathcal{V}^r$$

may be incorporated into the model formulation as a means of forbidding the assignment of certain MLE resources to certain bases at the end of their routes if they are never allowed to relocate to these bases. In addition, the distance patrol autonomy thresholds are incorporated in the formulation by means of the sets of linking constraints

$$-\left(a^d_{k\tau} - \sum_{\substack{i=0_{k\tau} \\ i \in \mathcal{V}_\tau^e}} \sum_{\substack{j \in \mathcal{V}_\tau^e \\ j \neq 0_{k\tau}}} d_{ijk\tau} x_{ijk\tau} - d_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau} x_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau} - A^d_{k\rho}\right) \leq A^d_{k\rho}(1 - w^d_{k\rho\tau})$$

and

$$x_{\ell\rho k\tau} \leq w^d_{k\rho\tau}, \quad \ell \in \mathcal{V}^e_\tau,\ k \in \mathcal{V}^r,\ \rho \in \mathcal{V}^p.$$

In the constraint sets above, MLE k is forbidden to be scheduled to a patrol circuit at the end of its route if its distance autonomy level at the end of the route, together with the distance to be travelled from the last VOI on that route to patrol circuit ρ, is below a pre-specified distance patrol autonomy threshold, where $w^d_{k\rho\tau}$ is a linking constraint variable. Similarly, the sets of linking constraints

$$-\left(a^t_{k\tau} - \sum_{\substack{i=0_{k\tau} \\ i \in \mathcal{V}_\tau^e}} \sum_{\substack{j \in \mathcal{V}_\tau^e \\ j \neq 0_{k\tau}}} \frac{d_{ijk\tau}}{\eta_k} x_{ijk\tau} - \frac{d_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau} x_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau}}{\eta_k} - A^t_{k\rho}\right) \leq A^t_{k\rho}(1 - w^t_{k\rho\tau})$$

and

$$x_{\ell\rho k\tau} \leq w^t_{k\rho\tau}, \quad \ell \in \mathcal{V}^e_\tau,\ k \in \mathcal{V}^r, \rho \in \mathcal{V}^p$$

ensure that MLE k is forbidden to be assigned to a patrol circuit at the end of its route if its time autonomy level at the end of the route is below a pre-specified time patrol autonomy threshold, where $w^t_{k\rho\tau}$ is also a linking constraint variable.

Finally, the constraints sets

$$\begin{aligned} w^d_{k\rho\tau}, w^t_{k\rho\tau} &\in \{0,1\}, \quad k \in \mathcal{V}^r,\ \rho \in \mathcal{V}^p, \\ x_{ijk\tau} &\in \{0,1\}, \quad i \in 0_{k\tau} \cup \mathcal{V}^e_\tau,\ j \in \mathcal{V}^e_\tau \backslash 0_{k\tau},\ k \in \mathcal{V}^r, \\ y_{ik\tau} &\in \{0,1\}, \quad i \in \mathcal{V}^e_\tau,\ k \in \mathcal{V}^r \end{aligned}$$

enforce the binary nature of the model variables.

Although not incorporated into the mathematical models of this chapter, it is important to point out that the feasible decision space is also expected to take into account the nature of the location of VOI intercepts. More specifically, it is acknowledged that there exist locations in space where servicing a VOI is not feasible. The maritime space beyond the boundaries of the jurisdiction area of a coastal nation will, for example, legally forbid certain MLE resources to service VOIs embodying certain types of threats[5], or certain MLE resources may not be

[5]Whether or not this constraint may be relaxed in real-life will typically depend on the severity of the threat and the international rights of the entity in control of the MLE resource.

allowed to operate inside certain maritime zones. Of course, this also includes the naive case of attempting to service a VOI with a calculated intercept located on land. The mathematical model of a given problem instance is therefore also expected to exclude solutions in which at least one such infeasible interception point exists from the feasible decision space.

4.3 Model configuration in an intermediate paradigm

In this section, a number of constituent elements are put forward for modelling an MLE response selection problem in an intermediate decision making paradigm (see §3.3.2). Because most of the elements in this paradigm overlap with the ones presented for the centralised paradigm, however, only previously unmentioned model elements are discussed in this section.

In addition to Objectives I–IV of the previous section, the following objective is proposed to inform the formulation of MLE response strategies in an intermediate decision making paradigm:

V Maximise the combined *consensus score* obtained by aggregating the level of agreement of each decision entity in respect of the set of VOIs assigned to it for MLE response selection operations, weighted by (a) an individual VOI preferential ordered set associated with each decision entity, (b) the deviation from the ideal quantity of VOIs assigned to each decision entity and (c) the relative importance of each decision entity to the coastal nation.

Following the discussion in §4.1.1, this fifth objective deals with measuring the consensus level achieved as a result of the assignment of VOIs to the various decision entities. It is assumed that this consensus level is measured by the weighted sum of the deviation from the decision entities' ideal or preferred subsets of VOIs to those in the solution. These ideal subsets may be constructed by selecting a certain number of elements from the preferential ordered set of VOIs of each decision entity. This objective may be expressed in two parts. First, a linear combination of the weighted scores of the level of VOI preference of decision entities and the VOIs distributed to them may be calculated. Then, a linear combination of penalties may be derived based on the deviation of the VOIs assigned to the decision entities from the ideal number of VOIs requested by these entities.

It is suggested that the penalties be assessed as a certain fixed discrete function of the deviations for three main reasons. First, a fixed function is necessary, in contrast to decision entity-based subjective functions, so as to ensure that the level of consensus is measured neutrally with respect to this objective, as there already exists a preferential weight associated with each decision entity (*i.e.* some entities may be deemed more important than others with respect to achieving the MLE goals of the coastal nation). Secondly, this function may be configured so as to reflect the relative importance of distributing VOIs with respect to the preferential ordered sets of the various decision entities to that of the deviation from the respective ideal subset sizes. Finally, such a function allows for asymmetry where, for example, assigning more VOIs to a decision entity than its ideal number is not penalised as strictly as assigning fewer VOIs.

Recall that $\mathcal{V}_\tau^e = \{1, \ldots, n_\tau\}$ is the set of VOIs observed at the beginning of time stage $\tau \in \mathbb{N}$, and let $\mathcal{Z}$ represent the set of decision entities of the coastal nation. Let $Z_s \in [0, 1]$ be a weight representing the relative importance of decision entity $s \in \mathcal{Z}$ to the coastal nation and define the ordered set $\mathcal{O}_{s\tau} = \{o_{s1}, \ldots, o_{sn_\tau}\}$ as the input data received from decision entity s with respect to VOI distribution preferences, so that VOIs are ranked from most preferred (*i.e.* most desirable to investigate) to least preferred, at the beginning of time stage τ. Also, define $N_{s\tau} \in \{0, 1, \ldots, n_\tau\}$ as the ideal number of VOIs that decision entity s would prefer to investigate during time stage

τ. Moreover, let f_c be a discrete function on the domain $\{-n_\tau, \ldots, n_\tau\}$ measuring the penalty incurred from a spread around the ideal number $N_{s\tau}$.

Furthermore, define the variable

$$z_{is\tau} = \begin{cases} 1, & \text{if VOI } i \text{ is scheduled on a route assigned to entity } s \text{ during time stage } \tau, \\ 0, & \text{otherwise.} \end{cases}$$

The aim of Objective V above is therefore to

$$\text{maximise} \quad \sum_{s \in \mathcal{Z}} Z_s \left[\sum_{i \in \mathcal{V}_\tau^e} z_{is\tau} \left(n_\tau - \mathcal{O}_{s\tau}(i) \right) - f_c \left(N_{s\tau} - \sum_{i \in \mathcal{V}_\tau^e} z_{is\tau} \right) \right],$$

where $\mathcal{O}_{s\tau}(i)$ is a cardinal number representing the position of VOI i in the set $\mathcal{O}_{s\tau}$ during time stage τ.

Finally, the assumption that scheduled routes do not intersect one another in the model configuration component of the DSS is also applicable to these new variables. Consequently, the constraint sets

$$\sum_{s \in \mathcal{Z}} z_{is\tau} \leq 1, \quad i \in \mathcal{V}_\tau^e$$

and

$$z_{is\tau} \in \{0, 1\}, \quad i \in \mathcal{V}_\tau^e, \; s \in \mathcal{Z}$$

have to be included in the model configuration within this decision making paradigm.

4.4 Model configuration in the decentralised paradigm

In this section, a number of constituent elements are finally put forward for modelling the MLE response selection problem in the decentralised decision making paradigm (see §3.3.3). Because most of the elements in this paradigm again overlap with the ones presented in respect of the other two decision making paradigms, only previously unmentioned model elements are discussed in this section.

As mentioned in §3.3.3, the VOI distribution operator and decision entity operators typically operate independently from one another and, consequently, the resulting routing problem instances occur independently from one another. Therefore, define a *global search* as the process of finding a solution to the VOI distribution problem and define a *global time stage* and a *global disturbance* in conjunction with this definition. Furthermore, define a *local search* as the process a decision entity undertakes to find a solution to the scheduling and routing problem involving its own MLE resources, and define a *local time stage* and a *local disturbance* in conjunction with this definition.

Because the routing process of an MLE response selection problem is subject to customer profits, entities are not obliged to schedule all VOIs assigned to them from the VOI distribution process to visitation routes. As a result, other entities may be interested in incorporating certain of these unscheduled VOIs in their routing process. The process of storing unscheduled VOIs is defined here as *VOI pooling*, and the *VOI pool* is stored and updated in the MLE database in real time. It is crucial that entity preferential ordered sets are not subject to changes as a result of this VOI pooling process — the opportunity of transferring a subset of unscheduled VOIs to another decision entity does not go through the VOI distribution process, as the problem is

resolved locally after being triggered by the local disturbance of the transfer. If more than one decision entity wishes to assume responsibility for a pooled VOI (*i.e.* incorporate the VOI in their own routing processes), it is the responsibility of the VOI distribution operator to break the tie by using some multi-person decision making process appropriate to the situation at hand.

As mentioned above, the decision entities are in charge of conducting parts of the overall MLE response selection routing process in their own, preferred ways. This incorporates, *inter alia*, the ability to decide which subset of the VOIs assigned to them should be scheduled for visitation (by employing the notion of customer visitation profits). It follows that no penalties should be incurred when the size of the subset of VOIs assigned to a decision entity at the beginning of a global time stage is larger than or equal to its ideal number of VOIs. A penalty should, however, be incurred if the size of this subset is smaller than the ideal number (in contrast to Objective V, where a penalty is incurred when the assigned number of VOIs is either smaller *or* larger than the ideal number). Objective VI is therefore specified analogously to Objective V above as

$$\text{maximise} \quad \sum_{s\in\mathcal{Z}} Z_s \left[\sum_{i\in\mathcal{V}_\tau^e} z_{is\tau} \left(n_\tau - \mathcal{O}_{s\tau}(i)\right) - f_c^+ \left(N_{s\tau} - \sum_{i\in\mathcal{V}_\tau^e} z_{is\tau} \right) \right],$$

where f_c^+ is a discrete function on the domain $\{-n_\tau, \ldots, n_\tau\}$ measuring the penalty incurred from the spread around the ideal VOI number $N_{s\tau}$, so that $f_c^+ = 0$ whenever $N_{s\tau} - \sum_{i\in\mathcal{V}_\tau^e} z_{is\tau} \leq 0$.

In contrast to an intermediate decision making paradigm, all VOIs are distributed among the decision entities. Consequently, the constraint sets

$$\sum_{s\in\mathcal{Z}} z_{is\tau} = 1, \qquad i \in \mathcal{V}_\tau^e$$

and

$$z_{is\tau} \in \{0,1\}, \qquad i \in \mathcal{V}_\tau^e,\ s \in \mathcal{Z}$$

are again incorporated into the model formulation within this decision making paradigm.

In addition to assigning tasks to the decision entities of the coastal nation in real time by achieving the maximum level of consensus during the VOI distribution process, the VOI distribution operator might also want to incorporate an objective function which measures the level of appropriateness between the decision entities and the nature of VOIs assigned to them (*i.e.* by considering the respective abilities of decision entities to neutralise certain types of threats, regardless of other preferential factors). In other words, this objective function would ensure that the appropriate VOIs are assigned to every decision entity. This implementation is, however, only justified if the VOI distribution operator is assumed to be a perfectly rational being with advanced knowledge in his field.

4.5 Chapter summary

The first step in solving the MLE response selection problem consists of selecting, populating and storing the constituent elements in the mathematical modelling subsystem, which resides in the model configuration component of the DSS architecture proposed in Chapter 3. In §4.1, underlying assumptions and concepts were discussed which are critical to modelling the MLE response selection problem. In particular, it was described how the problem may be modelled as a dynamic VRP. Fundamental model elements were also proposed for modelling the MLE response selection problem in a centralised, intermediate or decentralised decision making paradigm.

The model formulations presented in this chapter are of a fixed nature and reflect long-term conditions/preferences with respect to solving MLE response selection problem instances in real time. In the next chapter, certain dynamic modelling features are proposed which form part of the model management component. These features may be incorporated into the model formulation by the operator on a problem instance basis, so as to better reflect conditions/preferences with respect to a certain situation for the duration of one or more particular time stages.

CHAPTER 5

Model Management

Contents

In common with many routing and scheduling applications, the amount of information needed to define the problem is large. This information is more than just a list of rules and regulations and consists of both documented and undocumented knowledge, subject to both soft and hard interpretation which may be objective or subjective. It would appear at first sight that routing and scheduling is highly structured and well defined but the problem in practice is usually less clearcut. — R. Fink [38]

In an MLE response selection environment, it may be very difficult to satisfy all constraints and meet all goals when developing a set of routes, either manually or automatically, based on mathematical programming. A number of important features demonstrating the dynamism of the mathematical modelling process of the MLE response selection problem are presented in this chapter as a way of improving model flexibility to adapt to certain situations. More specifically, following a disturbance, or simply from a subjective preferential point of view, MLE operators may wish to provide additional input information to a problem instance in order to accommodate a variety of special requests into the model prior to launching the solution search process for the next time stage. The model management component provides operators with a

range of configuration modules that may be used to implement such requests and simultaneously reduce the computational complexity of the problem.

This chapter is structured as follows. Two main sub-components spanning the model management component are first introduced in §5.1, and this is followed by a more thorough description of each sub-component together with examples of the features they embody (in §5.2 and §5.3, respectively). A few comments are then made in §5.4 with respect to the incorporation of the model management component in a real-life MLE response selection DSS. The chapter finally closes with a brief summary in §5.5.

5.1 The model management sub-components

The integration of a model management component into the proposed MLE response selection DSS is motivated by two general reasons. First, it may be difficult for the solution search engine to generate a high-quality set of non-dominated solutions to a certain problem instance at the start of a time stage when the complexity of the problem is very high (as in most VRPs, this complexity is typically induced by the number of elements in the set of customers and the set of vehicles) or when the computational budget of the search process is subject to a strict time constraint. Secondly, the mathematical models proposed in Chapter 4 only include fundamental model structures; in reality, it may be inappropriate to assume that MLE response selection solutions sought by an operator will always reside within the feasible search space generated by these models. In other words, certain situations may arise at sea necessitating a certain degree of operator-induced subjectivity, which may be relatively unusual or infrequent, and hence not accounted for in the model configuration process.

The main reason for not attempting to incorporate every possible situation into the fundamental mathematical model structure is that the computational complexity of the problem would increase dramatically as a result, which is clearly undesirable when considering that these situations are expected to occur relatively infrequently. It is therefore critical to be able to differentiate between routine and unusual situations whilst configuring the mathematical modelling subsystem of the proposed DSS. Operator judgment may henceforth be incorporated as part of the MLE response selection input data in order to provide a more realistic and accurate picture of these unusual situations not accommodated in the fundamental model. Two sub-components of the model management component are proposed for this endeavour, namely a *cutting plane* sub-component (described in §5.2) and a *model adaptation* sub-component (described in §5.3), as depicted in Figure 5.1.

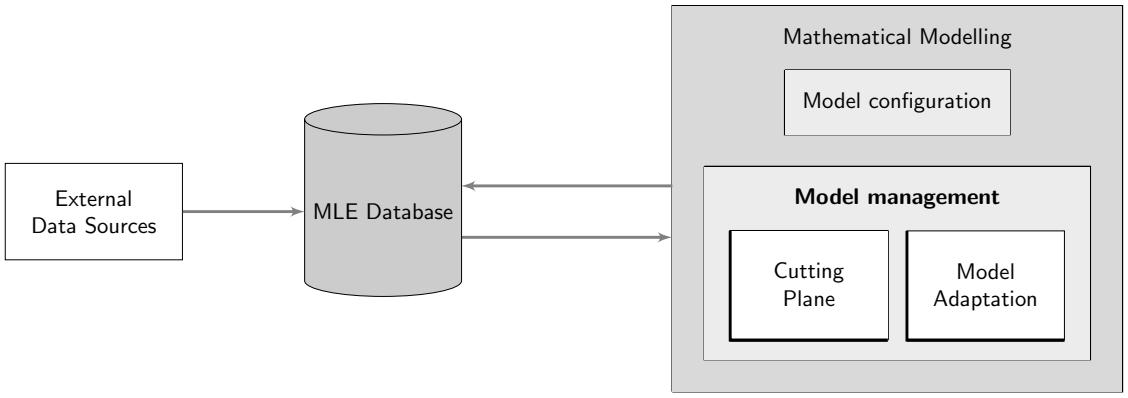


FIGURE 5.1: *The model management component with its two sub-components.*

5.2 The cutting plane sub-component

The purpose of the cutting plane sub-component is to allow for the construction of a faithful representation of a problem instance within the DSS which directs the solution search process in the desirable directions. This is achieved by incorporating operator expertise and knowledge in the field as part of the *a priori* input data, so that certain subsets of the decision space that the operator will, *a posteriori*, regard as undesirable in any case, do not form part of the feasible decision space. In other words, in addition to guiding the optimisation process towards generating high-quality solutions, the purpose of this sub-component is to contribute towards contracting the complexity of the problem instance, thereby increasing the likelihood of a better set of non-dominated solutions to be obtained within a tight computational budget. Reducing or *cutting* away portions of the search space in this manner empowers the cutting plane sub-component to tighten the model formulation by temporarily enforcing hard constraints in the formulation at the beginning of a time stage.

In this section, examples of features and concepts that may be included in the cutting plane sub-component are presented in some detail. In particular, the abilities to forbid certain MLE resources from visiting certain VOIs, to force certain VOIs to be allocated to routes visited by specific MLE resources, to associate end-of-route assignments with certain bases or patrol circuits, to dictate a lower bound of acceptability with respect to customer visitation profits and to impose assignments in the VOI distribution process (in the case of a decentralised decision making paradigm), are considered, and constraint sets are presented for use in such operator-induced configurations.

5.2.1 Exclusion sets for forbidden VOI assignments

Following the advent of certain disturbances, or deriving from purely subjective preference, an operator may wish to implement specifications which prohibit certain VOIs from being visited by certain MLE resources prior to launching the solution search process for the next time stage. In order to achieve such specifications, denote by $\mathcal{E}_{k\tau}$ the so-called *VOI exclusion set* associated with MLE resource k. This set contains those VOIs forbidden from being included in the visitation route of MLE resource k during time stage τ. It follows that $\mathcal{V}^e_{k\tau} \cap \mathcal{E}_{k\tau} = \emptyset$ for all $\tau \in \mathbb{N}$ and all $k \in \mathcal{V}^r$. In other words, any solution generated during the search process in which there is an interception route containing a VOI belonging to the exclusion set of the respective MLE resource is classified as infeasible.

To incorporate the above-mentioned visitation prohibition requirements into the model formulation, the set of constraints

$$\sum_{i \in \mathcal{E}_{k\tau}} y_{ik\tau} = 0, \qquad k \in \mathcal{V}^r$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ.

The reasons for employing VOI exclusion sets may vary from one operator to another. Cases where the visitation score (Objective I in §4.2.3) is not afforded much subjective preference during the solution analysis process, or the parameters contained in the particular objective function are not accurate, are examples of situations where an operator might opt for the specification of a VOI exclusion set. Also, in cases where the objective reflecting the risk of infeasible encounters (Objective IV in §4.2.3) is not selected as part of the fundamental model structure, the operator may wish to make use of VOI exclusion sets to prohibit the feasible region

of the decision space from containing routes along which certain MLE resources are scheduled to intercept certain VOIs associated with a relatively high risk of resulting in an infeasible encounter. If infeasible encounters were, however, to occur, the operator would clearly wish to forbid any MLE resources that are strictly incapable of neutralising the threat embodied by that VOI to be scheduled to a new interception route visiting this VOI in the future. Finally, in cases where a VOI travels at a significantly high velocity and is expected to reach the coast or escape from the jurisdiction area (unless a beyond-borders chase situation prevails) before the operator believes that it has a chance of being intercepted by any capable MLE resources, it should be possible to designate the VOI as a lost cause (*i.e.* this VOI should then be inserted into the VOI exclusion set of every MLE resource[1]).

5.2.2 Inclusion sets for imposed VOI assignments

In contrast to exclusion set requirements, the conditions imposed by *VOI inclusion sets* are met as long as the respective MLE resources *are* scheduled to intercept VOIs specified in these sets. In other words, any solution generated during the search process which contains a route omitting a VOI belonging to the inclusion set of an MLE resource is classified as infeasible. Additionally, the operator may prefer to implement an ordering in which VOIs belonging to certain inclusion sets should be visited so as to ensure that various degrees of urgency are taken into account during the scheduling of MLE resource visitation routes. It should, however, be noted that the visitation route of an MLE resource may, of course, also contain VOIs that are not part of its inclusion set. In this section, the use of three types of VOI inclusion sets is suggested, namely *unordered VOI inclusion sets* (VOIs in these sets are required to be intercepted by certain MLE resources, but with no particular degree of urgency in respect of the order in which these VOIs are visited), *ordered VOI inclusion sets* (VOIs in these sets are required to be intercepted by certain MLE resources in a particular order within their visitation routes) and *unordered VOI inclusion sets with ordered subsets* (VOIs in these sets are required to be intercepted by certain MLE resources, but only specified subsets of these VOIs are required to be intercepted in a particular order within the MLE resource visitation routes).

The reasons for employing VOI inclusion sets may again vary from one operator to another. Examples of situations necessitating the use of such inclusion sets include some of those discussed in the previous section for VOI exclusion sets. A VOI being close to the boundaries of the jurisdiction area or close to the shore of a coastal nation, for example, should be placed in the first position of the VOI ordered inclusion set of a capable MLE resource if it is deemed to exhibit a sufficiently high level of threat. Also, if an MLE resource has not completed servicing a VOI at the end of a time stage, or is relatively close to some VOI it was scheduled to visit at the end of the time stage, it may be wise to include this VOI in the inclusion set of the MLE resource at the beginning of the next time stage, particularly when the delay score (Objective II in §4.2.3) is not highly prioritised in the decision making process.

Unordered VOI inclusion sets

As described above, unordered VOI inclusion sets may be used to force certain VOIs to be intercepted by certain MLE resources, but with no particular degree of urgency in respect of the order in which these VOIs are visited. In other words, the conditions imposed by such sets are satisfied as long as the respective MLE resources are scheduled to intercept VOIs placed in these sets.

[1] Alternatively, this VOI may simply be removed manually from the set $\mathcal{V}_\tau^e$.

Define the unordered VOI inclusion set $\mathcal{I}_{k\tau}$ to contain those VOIs required to be included in the visitation route of MLE resource k during time stage τ. It is assumed that $\mathcal{I}_{k\tau} \cap \mathcal{I}_{\ell\tau} = \emptyset$ for all $\tau \in \mathbb{N}$ and all $k, \ell \in \mathcal{V}^r$, with $k \neq \ell$. Furthermore, $\mathcal{I}_{k\tau} \subseteq \mathcal{V}^e_{k\tau}$ for all $\tau \in \mathbb{N}$ and $k \in \mathcal{V}^r$. To incorporate the above-mentioned visitation requirements into the model formulation, the set of constraints

$$\sum_{\iota \in \mathcal{I}_{k\tau}} y_{\iota k\tau} = |\mathcal{I}_{k\tau}|, \qquad k \in \mathcal{V}^r$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ. The notion of an unordered VOI inclusion set for MLE resource k during time stage τ set is illustrated graphically in Figure 5.2.

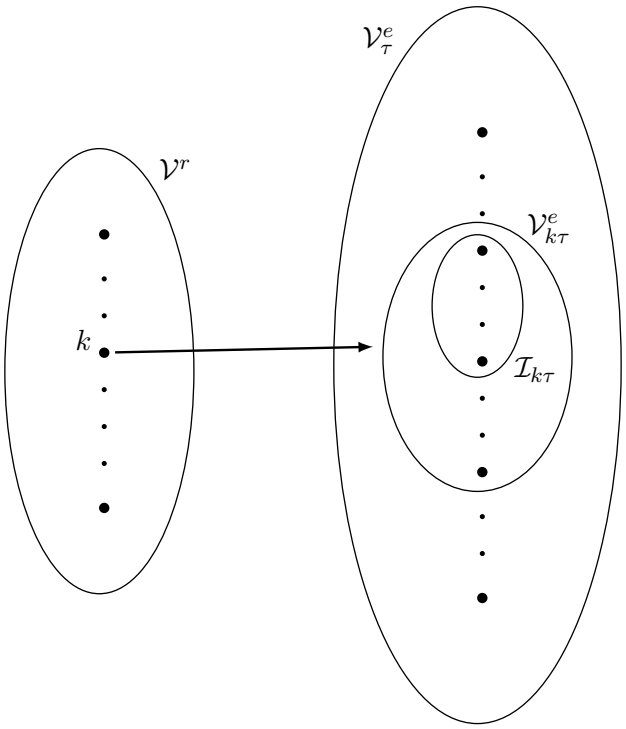


FIGURE 5.2: *Unordered VOI inclusion set of MLE resource k during time stage τ.*

Ordered VOI inclusion sets

As described earlier, VOIs in the ordered VOI inclusion set of an MLE resource are required to be intercepted by that MLE resource in a particular order within its visitation route. Such a set may be employed, for example, when the operator does not believe that the search process will yield visitation routes for particular MLE resources in the most desirable order possible (see Objective II for the scheduling process), so as to respect a certain degree of urgency in the visitation order. In order to incorporate this degree of urgency efficiently in the model formulation, it is assumed that the MLE resource will first visit the VOIs belonging to its ordered inclusion set before visiting any other VOIs assigned to it within its route. Moreover, it is assumed that the order in which VOIs are entered in such a set is the order in which the MLE resource should visit them within its route (*i.e.* the first element in an ordered inclusion set represents the first VOI visited by the respective MLE resource within its route, the second element is the second VOI to be visited, *etc*).

Hence, define the ordered VOI inclusion set $\widehat{\mathcal{I}}_{k\tau} \subseteq \mathcal{V}^e_{k\tau}$ to contain those VOIs required to be included in the visitation route of MLE resource k during time stage τ, so that $\widehat{\mathcal{I}}_{k\tau}(\iota) = \mathcal{V}^e_{k\tau}(\iota)$ for all $\iota \in \{1, \ldots, |\widehat{\mathcal{I}}_{k\tau}|\}$ and all $\tau \in \mathbb{N}$. To incorporate the above-mentioned visitation requirements into the model formulation, the sets of constraints

$$\begin{aligned} x_{0_k(\widehat{\mathcal{I}}_{k\tau}(1))k\tau} &= 1, && k \in \mathcal{V}^r \\ \text{and} \quad \sum_{\iota \in \{1,\ldots,|\widehat{\mathcal{I}}_{k\tau}|-1\}} x_{(\widehat{\mathcal{I}}_{k\tau}(\iota))(\widehat{\mathcal{I}}_{k\tau}(\iota+1))k\tau} &= |\widehat{\mathcal{I}}_{k\tau}| - 1, && k \in \mathcal{V}^r,\ |\widehat{\mathcal{I}}_{k\tau}| > 1 \end{aligned}$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ.

Unordered VOI inclusion sets with ordered subsets

Including unordered VOI inclusion sets with ordered subsets in the model formulation allows the operator to request that an MLE resource be scheduled to intercept a specific set of VOIs within its visitation route, in such a way that a certain subset of this set of VOIs carries an immediate degree of urgency and are required to be visited first and in a particular order, while the remaining VOIs are required to be visited later by the same MLE resource, but in no particular order. Such an inclusion set therefore combines the concepts of unordered VOI inclusion sets and ordered VOI inclusion sets presented above. Employing this type of VOI inclusion set allows the operator to prioritise the order in which certain VOIs are visited by a particular MLE resource while being indifferent with respect to the order in which the remaining VOIs are visited.

Hence, define the ordered VOI inclusion set $\tilde{\mathcal{I}}^o_{k\tau} \subseteq \tilde{\mathcal{I}}_{k\tau}$ to contain those VOIs required to be included first in the specified order within the visitation route of MLE resource k during time stage τ, while $\tilde{\mathcal{I}}^u_{k\tau} = \tilde{\mathcal{I}}_{k\tau} \backslash \tilde{\mathcal{I}}^o_{k\tau}$ contains the VOIs required to be visited later by the same MLE resource, but in no particular order. To incorporate the above-mentioned visitation requirements into the model formulation, the sets of constraints

$$\begin{aligned} x_{0_k(\tilde{\mathcal{I}}^o_{k\tau}(1))k\tau} &= 1, && k \in \mathcal{V}^r, \\ \sum_{\iota \in \{1,\ldots,|\tilde{\mathcal{I}}^o_{k\tau}|-1\}} x_{(\tilde{\mathcal{I}}^o_{k\tau}(\iota))(\tilde{\mathcal{I}}^o_{k\tau}(\iota+1))k\tau} &= |\tilde{\mathcal{I}}^o_{k\tau}| - 1, && k \in \mathcal{V}^r,\ |\tilde{\mathcal{I}}^o_{k\tau}| > 1 \\ \text{and} \quad \sum_{\iota \in \tilde{\mathcal{I}}^u_{k\tau}} y_{\iota k\tau} &= |\tilde{\mathcal{I}}^u_{k\tau}|, && k \in \mathcal{V}^r \end{aligned}$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ.

This particular type of VOI inclusion set specification is a generalisation of the formulations proposed in the other two approaches above. In particular, the case in which $\tilde{\mathcal{I}}^u_{k\tau} = \tilde{\mathcal{I}}_{k\tau}$ reduces to the specification of the unordered VOI inclusion set $\mathcal{I}_{k\tau}$, while the case in which $\tilde{\mathcal{I}}^o_{k\tau} = \tilde{\mathcal{I}}_{k\tau}$ reduces to the specification of the ordered VOI inclusion set $\widehat{\mathcal{I}}_{k\tau}$.

5.2.3 End-of-route assignments

In Chapter 4, distance and time autonomy thresholds were introduced as part of the model formulation for the purpose of stipulating that an MLE resource is not allowed to end its route

by being assigned to a certain patrol circuit if its distance or time autonomy level after having visited the last VOI on its visitation route is below a certain level. Additionally, a similar base assignment restriction was also incorporated in the formulation, in the sense that certain MLE resources may not be allocated to end their visitation routes at certain bases. These conditions are, however, fixed and are therefore always included in the model formulation (by definition of the model configuration component).

This section is devoted to a discussion on the last arc of every route, where active MLE resources transfer from an active state to an idle state in a sub-process defined as *end-of-route* or *post-mission* assignments. A major aspect of the MLE response selection problem involves assignment decisions for implementation after visitation route completion, which consists of determining where certain MLE resources should or should not be sent after completing their missions. For example, the idle MLE resources management operator may wish to distribute idle MLE resources amongst the bases by managing their spread and strategic placement.

Information pertaining to end-of-route assignment preferences ought to be communicated to the response selection DSS at the beginning of every time stage. This feature of the MLE response selection problem therefore requires a certain form of input from the idle MLE resources DSS[2]. After having intercepted the VOIs assigned to it (*i.e.* after having completed its mission), the idle MLE resources management operator may either assign an MLE resource to travel to a designated base or to join a designated patrol circuit. Because the MLE response selection process determines where MLE resources will be located in space after completing their missions, it is crucial to consider the impact of this final route arc with respect to operating costs and distance and time-constrained feasibility. Ultimately, a route may only be classified as distance or time-constrained feasible if there exists at least one approved base that is at most as far away as the most constraining autonomy level threshold associated with the MLE resource immediately after having intercepted and serviced the last VOI on its route.

Dictating end-of-route assignments may be achieved in a manner similar to forcing or prohibiting VOI visitations by certain MLE resources, but this type of specification is instead represented by sets containing one or more elements from the patrol circuit and MLE resource base sets. It is assumed that a set $\mathcal{V}^b_{k\tau} \subseteq \mathcal{V}^b$ containing the set of bases to which MLE resource k may travel after completing its mission during time stage τ, is issued as input data received from the idle MLE resources management DSS. Similarly, let $\mathcal{V}^p_{k\tau} \subseteq \mathcal{V}^p$ contain the set of patrol circuits to which MLE resource k may be assigned after completing its mission during time stage τ (independently from the patrol autonomy thresholds constraints presented in §4.2.4).

Implementing so-called *forbidden end-of-route base assignments* may be achieved by prohibiting MLE resource k from being scheduled to end its route at a base $b \in \mathcal{V}^b \backslash \mathcal{V}^b_{k\tau}$ during time stage τ. The set of constraints

$$\sum_{i \in \mathcal{V}^e_\tau} \sum_{\substack{b \in \mathcal{V}^b \\ b \notin \mathcal{V}^b_{k\tau}}} x_{ibk\tau} = 0, \qquad k \in \mathcal{V}^r$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ in order to avoid such assignments. Similarly, implementing so-called *imposed end-of-route base assignments* may be achieved by requiring MLE resource k to be scheduled to end its route at any one of the bases in $\mathcal{V}^b_{k\tau}$ during time stage τ. The set of constraints

$$\sum_{i \in \mathcal{V}^e_\tau} \sum_{b \in \mathcal{V}^b_{k\tau}} x_{ibk\tau} = 1, \qquad k \in \mathcal{V}^r$$

[2]The DSS in support of the allocation of idle MLE resources in both time and space is external to the MLE response selection DSS (see §3.1).

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ in order to accommodate assignments of this kind. This notion is illustrated in Figure 5.3, where MLE resource k is scheduled to end its route at an approved base $b \in \mathcal{V}^b_{k\tau}$ after having visited the VOIs in $\mathcal{V}^e_{k\tau}$.

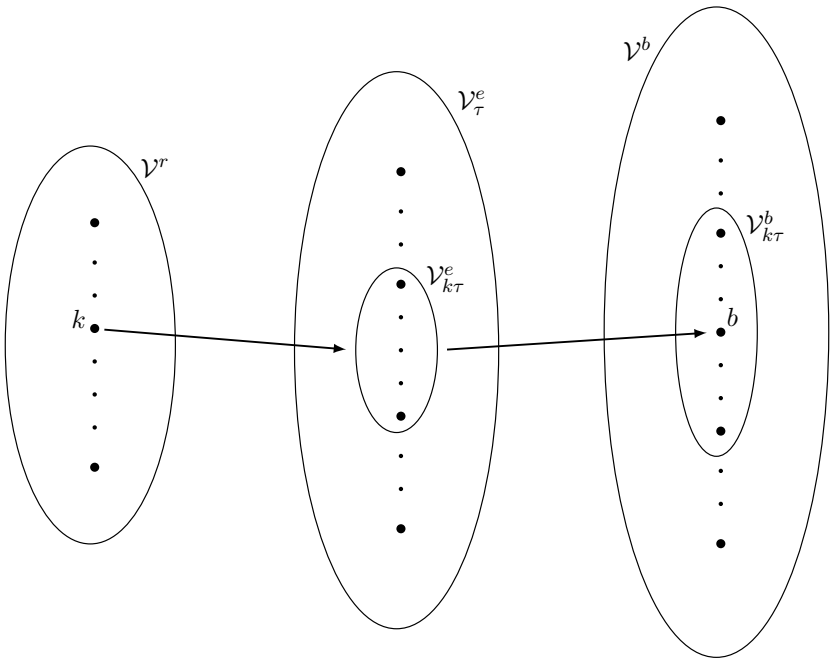


FIGURE 5.3: *Imposed end-of-route base assignment for MLE resource k during time stage τ.*

Implementing so-called *forbidden end-of-route patrol assignments* may be achieved by prohibiting MLE resource k from being scheduled to end its route at a patrol circuit $\rho \in \mathcal{V}^p \backslash \mathcal{V}^p_{k\tau}$ during time stage τ. The set of constraints

$$\sum_{i \in \mathcal{V}^e_\tau} \sum_{\substack{\rho \in \mathcal{V}^p \\ \rho \notin \mathcal{V}^p_{k\tau}}} x_{i\rho k\tau} = 0, \qquad k \in \mathcal{V}^r$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ in order to avoid such assignments. Finally, implementing so-called *imposed end-of-route patrol assignments* may be achieved by requiring MLE resource k to be scheduled to end its route at any one of the patrol circuits in $\mathcal{V}^p_{k\tau}$ during time stage τ. The set of constraints

$$\sum_{i \in \mathcal{V}^e_\tau} \sum_{\rho \in \mathcal{V}^p_{k\tau}} x_{i\rho k\tau} = 1, \qquad k \in \mathcal{V}^r$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ in order to accommodate assignments of this kind.

In any case, however, it should be noted that the idle MLE resources management operator has the power to decide on the destination of MLE resources as soon as they complete their missions, since they then transition to an idle state. Consequently, these idle MLE resources do not have to follow the pre-scheduled end-of-route assignments as dictated by the MLE response selection solution uncovered during the search process at the beginning of the time stage, because end-of-route preferences may in the meantime have evolved differently over time. Furthermore, it

is critical for both resource assignment DSSs to work together when configuring these end-of-route assignments; if the idle MLE resources manager is too strict with respect to end-of-route destination decisions, it may prevent very good routes from being scheduled, as certain active MLE resources may be restricted to travel within unnecessarily constrained zones. Hence, it is the duty of the idle MLE resource manager to ensure that MLE response selection operations do not suffer significantly as a result of poorly configured end-of-route assignment input data.

5.2.4 Customer profit thresholds

As was mentioned in Characteristic 2 of §4.1.2, any subset of VOIs may be serviced during any time stage. The search process is therefore configured to explore a possibly large variety of subsets in solution space in order to generate an acceptable spread of non-dominated solutions. The operator may, however, be inclined to focus his preferences on solutions with high visitation scores *a priori* (see Objective I in §4.2.3). Because a good upper bound on the visitation score is known in advance, however, it may be impractical to reduce the search space according to a pre-determined numerical score in this respect. Since the number of VOIs visited in a solution positively correlates to its visitation score, and because the number of VOIs at the beginning of a time stage is known, the operator may nevertheless be able to specify a threshold in the form of a minimum proportion of all VOIs that should be intercepted during the current time stage[3]. Such a threshold should ensure that the search process does not waste resources investigating solutions in which the subset of intercepted VOIs is deemed too small.

Hence, define a so-called *customer profits threshold* parameter $\Delta_\tau \in [0, 1]$ as the minimum proportion of VOIs that have to be intercepted during time stage τ; *i.e.* solutions that do not satisfy the inequality

$$\left| \bigcup_{k \in \mathcal{V}^r} \mathcal{V}^e_{k\tau} \right| \geq n_\tau \Delta_\tau$$

are classified as infeasible. In order to implement this requirement, the set of constraints

$$\sum_{i \in \mathcal{V}^e_\tau} \sum_{k \in \mathcal{V}^r} y_{ik\tau} \geq n_\tau \Delta_\tau$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ.

5.2.5 VOI distribution

Only model management features for use in the MLE response selection routing process have been proposed up to this point. Additionally, the VOI distribution model presented in §4.4 may also be subject to dynamic elements, as requested by the VOI distribution operator (provided that he has the power to do so). In particular, he may decide *a priori* to assign one or more specific VOIs to certain decision entities[4].

Hence, define the *VOI distribution inclusion set* $\mathcal{I}^s_\tau$ to contain those VOIs required to be allocated to decision entity s during time stage τ, noting that $\mathcal{I}^s_\tau \subseteq \mathcal{V}^e_\tau$ for all $\tau \in \mathbb{N}$ and all $s \in \mathcal{Z}$. In

[3] If the operator most often exhibits the same preferences, then these conditions may be incorporated as part of the model configuration component.

[4] The notion of prohibiting certain VOIs from being assigned to certain decision entities appears to be unrealistic and is thus omitted from the discussion.

order to incorporate the above-mentioned distribution requirements, the set of constraints

$$\sum_{i \in \mathcal{I}_\tau^s} z_{is\tau} = |\mathcal{I}_\tau^s|, \qquad s \in \mathcal{Z}$$

may (temporarily) be included in the VOI distribution process at the beginning of time stage τ.

5.3 The model adaptation sub-component

As mentioned in §5.1, the fundamental (fixed) model features included in the model configuration component cannot always accommodate all types of MLE response selection situations or necessarily conform to operator judgment or preferences. In this section, a number of examples of such situations, as well as ways of representing them in the mathematical modelling process, are briefly described. Unlike the features found in the cutting plane sub-component, features of the model adaptation sub-component are not necessarily configured as a function of time stages. In other words, the operator may sometimes wish to implement changes in the model formulation independently of the occurrence of disturbances. The features of this sub-component are typically achieved by a temporary constraint relaxation, a specification of temporary hard constraints or a modification of MLE response selection sets.

5.3.1 Combined assignments

It has been assumed up to this point that a VOI may only be scheduled for interception by at most one MLE resource. As stated in Characteristic 10 of §4.1.2, however, certain threatening MLE situations may require the use of multiple MLE resources in order to successfully neutralise them. Henceforth, define the notion of a *combined assignment* as the strategic unification of multiple MLE resources with respect to the interception of one or more specific VOIs. Combined assignments may be implemented whenever a strong requirement is present to neutralise the threats embodied in certain VOIs as efficiently as possible (in order, for instance, to reduce the risk of infeasible encounters or reduce the expected service times of these VOIs). Two examples of combined assignments are presented in this section, namely *converged assignments*, where multiple MLE resources are scheduled to intercept a single VOI for a once-off event, and *convoyed assignments*, where multiple MLE resources are scheduled to travel along a visitation route together. Combined assignments are assumed to be capable of neutralising a threat more effectively than single MLE resources, contributing towards preventing the risk of infeasible encounters, improving counter-threat performances and delivering shorter service times, as the converging subset combines the manpower and other resources from several MLE resources.

Converged assignments

Converged assignments deal with once-off events that are typically deemed very threatening, and require the use of multiple MLE resources in order to increase the chances of successfully neutralising the threat involved. This process is conducted in a "drop everything you are doing" attitude where, following the start of the time stage triggered by a VOI, the operator allocates a subset of MLE resources to converge on that VOI from their current locations. Each of these MLE resources is therefore re-scheduled along a route containing a single, common VOI; that is, they are immediately sent to the VOI involved without visiting any other VOIs previously assigned to them. Interestingly, if an operator would, for instance, prefer that an MLE resource

completes servicing a VOI it may currently be busy investigating, intercept another specific VOI prior to investigating the involved VOI or investigate other VOIs after investigating the involved VOI, then he may, in addition to the sets presented in this section, make use of the sets described in §5.2.2.

Suppose $i^* \in \mathcal{V}^e_\tau$ is a VOI which causes the operator to subjectively trigger a disturbance requesting a converged assignment situation, and define the set $\mathcal{V}^r_{i^*\tau} \subseteq \mathcal{V}^r$ to contain the subset of MLE resources scheduled by the operator to converge to VOI i^* during time stage τ. By definition, it follows that $\mathcal{V}^e_{k\tau} = \{i^*\}$ for all $k \in \mathcal{V}^r_{i^*\tau}$. The basic idea behind the concept of converged assignments in MLE response selection operations is illustrated graphically in Figure 5.4.

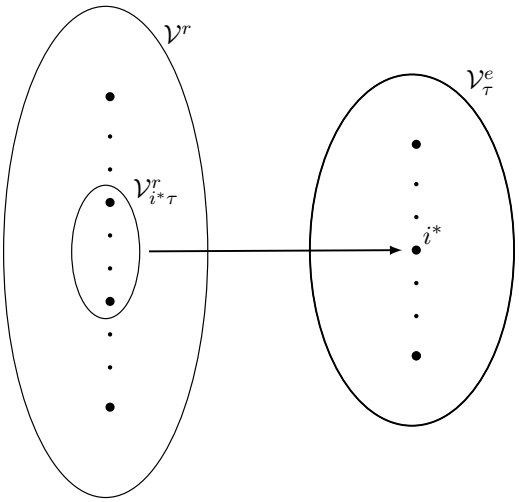


FIGURE 5.4: *Graphical representation of a converged assignment scenario.*

In order to incorporate the above-mentioned visitation requirements, the set of constraints

$$x_{0_k i^* k\tau} = 1, \qquad k \in \mathcal{V}^r_{i^*\tau}$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ. In addition, the constraint set (4.2.4) in §4.2.4, which states that no more than one MLE resource may visit a VOI, should temporarily be relaxed at the beginning of time stage τ, replacing it by

$$\sum_{\substack{k \in \mathcal{V}^r \\ k \notin \mathcal{V}^r_{i^*\tau}}} y_{ik\tau} \;\leq\; 1, \qquad i \in \mathcal{V}^e_\tau.$$

In cases of more than one converged assignment during any given time stage, it is assumed that each such assignment involves its own VOIs and subset of MLE resources; that is, if $i^*, j^* \in \mathcal{V}^e_\tau,\ j^* \neq i^*$ are VOIs involved in respective situations of converged assignments, then $\mathcal{V}^r_{i^*\tau} \cap \mathcal{V}^r_{j^*\tau} = \emptyset$.

Convoyed assignments

In convoyed assignments, multiple MLE resources are required to travel together along a route induced by one or more VOIs. These MLE resources are required to begin their route at a

pre-specified common location, and together intercept the VOIs assigned to them. One way of implementing such a requirement is to include certain instance constraints in the model formulation and redefine certain fundamental constraints and parameters. Undertaking this task is, however, not trivial, and is expected to add considerable complexity to the model formulation. A simpler and more efficient approach is to redefine the set of MLE resources $\mathcal{V}^r$ so as to temporarily combine a certain subset of MLE resources as one entry.

Hence, define the set $\mathcal{V}^r_{c\tau} = \{1, \ldots, |\mathcal{V}^r_{c\tau}|\} \subseteq \mathcal{V}^r$ to contain MLE resources that the operator wishes to configure in a convoyed assignment during time stage τ, and redefine the fleet set

$$\mathcal{V}^r = \begin{cases} \mathcal{V}^r_\tau = \{1, \ldots, \mathcal{V}^r_{c\tau}, \ldots, m + 1 - |\mathcal{V}^r_{c\tau}|\}, & \text{if } |\mathcal{V}^r_{c\tau}| \leq m - 2 \\ \mathcal{V}^r_\tau = \{1, \mathcal{V}^r_{c\tau}\}, & \text{if } |\mathcal{V}^r_{c\tau}| = m - 1 \\ \mathcal{V}^r_\tau = \mathcal{V}^r_{c\tau}, & \text{if } |\mathcal{V}^r_{c\tau}| = m \end{cases}$$

for this situation as the updated set of MLE resources during time stage τ. Note that, in this case, although $\mathcal{V}^r$ and $\mathcal{V}^r_\tau$ are formulated differently, they have the exact same composition of MLE resources. Furthermore, the MLE resources involved in such a set are assumed to be in an idle state while they converge towards one another at the start of their route (some pre-defined point in space), and the disturbance is triggered as soon as these MLE resources are grouped and ready to set out along their convoy route.

More than one case of convoyed assignments may occur during any given time stage, in which case it is assumed that each involves its own subset of MLE resources and visits its own subset of VOIs as in any of the other visitation routes); that is, if $\mathcal{V}^r_{c_1\tau} \subseteq \mathcal{V}^r$ and $\mathcal{V}^r_{c_2\tau} \subseteq \mathcal{V}^r$ are two MLE resource sets used in convoyed assignments during time stage τ, then $\mathcal{V}^r_{c_1\tau} \cap \mathcal{V}^r_{c_2\tau} = \emptyset$.

Possible implementation

The major challenge associated with combined assignments lies in (rapidly) reassessing certain critical fixed and dynamic parameters at the beginning of the time stage. For example, the expected service time of the VOI involved in a converged assignment scenario has to be evaluated in case one of the MLE resources involved is assigned to a route containing another VOI to be visited after having intercepted and serviced the VOI involved in the converged assignment, as the delay time associated with this other VOI has to be evaluated during the solution search process. Moreover, the counter-threat performances, expected service times, and travel speeds of a group of MLE resources configured to the same route in a convoyed assignment process should also be re-evaluated accordingly, which is certainly not a simple task.

5.3.2 Non-operational MLE resources

In some cases, an active MLE resource may experience technical problems while on a mission, causing it no longer to be able to carry on with its visitation route[5]. Active MLE resources experiencing such technical problems while in an active state are then classified as *non-operational*. The current problem instance must then be resolved while prohibiting the problematic MLE resource from operating in an active state. Note that in the case where an idle MLE resource in $\mathcal{V}^r$ were to experience such technical problems, it would not affect the current MLE response

[5]Such a situation is not to be confused with MLE resources undergoing maintenance or repairs; these MLE resources are currently not part of the set $\mathcal{V}^r$ and are, by definition, therefore neither active nor idle, but *unavailable*.

selection process, and it is the responsibility of the idle MLE resources DSS to deal with it accordingly.

One way of accommodating this scenario is to redefine the set of MLE resources by excluding the non-operational MLE resource, say $\overline{k}$, from the set of MLE resources $\mathcal{V}_\tau^r$, by performing the substitution $\mathcal{V}_\tau^r \leftarrow \mathcal{V}_\tau^r \backslash \overline{k}$, until the MLE resource is functional again. Another way of accommodating this feature is simply to include the constraint

$$\sum_{j \in \mathcal{V}_\tau^e} x_{0_{\overline{k}} j \overline{k} \tau} = 0$$

in the mathematical modelling process at the beginning of every time stage τ (while maintaining $\overline{k} \in \mathcal{V}_\tau^r$), from the time that the MLE resource is classified as non-operational until it is functional again.

5.3.3 Capacity restrictions

In certain situations, it may be the case that an MLE resource is no longer able to continue along its initial route after having intercepted one or more VOIs embodying certain types of threats, and is subsequently required to return to a base prior to servicing any further VOIs. Situations of this type are referred to as *capacity restrictions*. Such situations may not necessarily be predictable in advance as the natures of VOI interception (even ones known in advance and embodying the same type of threat) may differ significantly from one to another. Suppose that an MLE resource, say $\tilde{k}$, is experiencing a capacity restriction situation. Then, the constraint

$$\sum_{b \in \mathcal{V}^b} x_{0_{\tilde{k}} b \tilde{k} \tau} = 1$$

may (temporarily) be included in the mathematical modelling process at the beginning of time stage τ in order to accommodate this situation.

5.3.4 Deployable bases

It has been assumed, up to this point, that the set of bases is fixed (or at least updated independently of MLE response selection operations, such as, for example, as a result of the construction of a new coastal base) and that their locations are also fixed in space. Some coastal nations may, however, possess *deployable bases* (See Characteristic 15 in §4.1.2), typically in the form of very large vessels with very high autonomy levels that may strategically be moved in space. Areas at sea that are relatively far away from the coastline and/or in which a relatively high concentration of VOIs are known to appear are examples of strategic candidate locations for deployable bases. These bases provide the same basic services to a certain subset of MLE resources as coastal bases do between missions, such as fuel replenishment, basic maintenance, crew shift changes and possibly stationing). Deployable bases may be treated like any other bases in $\mathcal{V}^b$, with the exception that the number of MLE resources allowed to use such facilities is typically more restricted.

Due to their mobility, however, deployable base locations have to be updated at the beginning of every time stage. Moreover, if a deployable base has significantly moved in space over a relatively long, disturbance-free period of time, an operator may trigger a new time stage in order to resolve the current MLE response selection problem, as certain routes may no longer be distance or time feasible (particularly if the distance from an MLE resource to its pre-assigned end-of-route deployable base has increased significantly).

5.4 Comments

As mentioned in §4.1.10, as well as previously in this chapter, the set of MLE resources $\mathcal{V}^r$ is assumed not to be updated according to MLE response selection time stages. It must, however, be acknowledged that, while in an idle state, an MLE resource may not always be available for a mission in the near future. In particular, if an MLE resource is undergoing some form of maintenance, it would be impossible to consider it for a mission. It is therefore the responsibility of the MLE response selection or idle MLE resources management operator to update the set of MLE resources capable of undergoing a mission, in real time. Furthermore, in the case of deployable bases, as described in §5.3.4, it is also the duty of a MLE resource assignment operator to ensure that the geographical locations of such bases are updated in real time. Similar requirements also apply to patrol circuits, which may frequently be reconfigured in the idle MLE resources DSS.

In addition, because model management features may be large in quantity and/or overwhelming to reconfigure at the beginning of every time stage, it may be more practical to automatically carry over all model management configuration constraints from one time stage to another, except for those constraints (perhaps marked by an operator) not to be considered in force anymore. For example, if a VOI was placed in the VOI inclusion set of an MLE resource at the beginning of the previous time stage, and that VOI has since been intercepted and its service completed during that time stage, then it is obvious that the VOI in question should be removed from the VOI inclusion set of the respective MLE resource at the beginning of the next time stage.

Finally, if the model management component proposed in this chapter is incorporated into a real-life MLE response selection DSS, it should also include an additional sub-component, automated to alert the operator of possible inconsistencies or human errors whilst manually configuring features in the other two sub-components. For example, it is easy to see that the unordered VOI inclusion of a certain MLE resource may not intersect with the VOI exclusion set or the ordered VOI inclusion set of the same MLE resource. In addition, this automated sub-component should also be able to clear or update certain model management features from one time stage to another (as discussed above) on its own.

5.5 Chapter summary

This chapter contained an overview of some important dynamic features and constraints that may be incorporated in the mathematical modelling process in order to accommodate a variety of special requests or instructions prior to launching the solution search process for the next time stage, as configured by the operator.

The two sub-components of the model management component which are responsible for the inclusion of these constraints were first introduced in §5.1. These sub-components are the cutting plane sub-component, responsible for directing the solution search process in desirable directions (by incorporating operator expertise as part of the input data), and the model adaptation sub-component, in which situations that are not accounted for in the model configuration may be incorporated in the formulation of the problem for a particular problem instance.

The first sub-component above was then described in more detail in §5.2. In particular, the notions of VOI exclusion sets, VOI inclusion sets and end-of-route assignments were introduced, among other topics. The second sub-component was subsequently described in more detail in §5.3, where the notion of combined assignments was introduced, among other topics. A number

of general comments were finally made in §5.4 in respect of incorporating the model management component in a real MLE response selection DSS.

The final step involved in the design of an MLE response selection model formulation is to configure an array of various MLE response selection stochastic mathematical model components capable of managing the uncertainty evolving around the impact of non-deterministic input data on the development of a solution in real-time. In particular, various analyses are necessary in order to assess the extent to which a theoretical, *a priori* solution generated by the MLE response selection DSS at the start of a problem instance is expected to differ from the corresponding, *a posteriori* turn of events manifesting themselves in the process of implementing this solution in reality. This is the topic under consideration in the following chapter.

CHAPTER 6

Management of stochastic information

Contents

No mathematical model should be labeled realistic without at least considering the various facets of uncertainty in the decision and objective spaces of the underlying model. In other words, one may argue that deterministic models do not adequately describe a variety of situations in cases where the nature of some subset of input data is not known with certainty. Data of this kind are referred to as *stochastic*.

Although the expected values of stochastic input data were considered in the deterministic MLE response selection models put forward in Chapter 4, the variational spread around these values was not accounted for, and it is strongly anticipated that such uncertainty cannot assumed to be insignificant from the view point of an operator or decision maker. Instead, it is suggested that stochastic input data rather be modelled as probability distributions in which the level of uncertainty around the expected values of the underlying parameters is expressed by means of numerical variances.

The goal in this chapter is to design an array of stochastic mathematical model components, which may be incorporated in the MLE response selection models of Chapter 4, as a means to allow an operator to manage the risk of uncertainty associated with the various stochastic elements of this problem. Practically speaking, an operator may wish to be aware of the expected level of extent to which the *a posteriori* and real-time progression of a particular MLE response selection solution differs from its *a priori* evaluation in both decision space and objective space.

This chapter is structured as follows. The MLE response selection problem is first described in some detail within a stochastic paradigm in §6.1. This is followed in §6.2 by an investigation on effective ways in which to model the uncertainty pertaining to the position of VOIs in space. A step-by-step construction of model components that incorporate the uncertainty associated with VOI visitation locations follows in §6.3, while §6.4 contains a formulation of the stochasticity of

route travel distances. In §6.5, a brief discussion on the uncertainty pertaining to VOI service times is conducted, after which the management of uncertainty pertaining to the threatening nature of VOIs is addressed in §6.6. The focus then shifts in §6.7 from the impact that stochastic elements have on the feasible decision space to the impact they have on the evaluation of solution performances in objective space. The chapter finally closes with a brief summary in §6.8.

6.1 The stochastic MLE response selection problem

All models are wrong, but some models are useful. — G. Box [11]

Although the mathematical models derived in Chapter 4 may seem potentially useful, it is foreseen that the progression of the visitation routes over the course of a given time stage is very unlikely to turn out almost exactly as planned *a priori*. In fact, it is critical to acknowledge, for the sake of realism, that the level of unpredictability in the development of visitation routes in most real-life scenarios may even be so high as to not give the MLE response selection DSS presented in this dissertation any edge over human operators in addition to validating their decisions.

For this reason, it is both advocated and stressed that the level of practicality and realism in the *a priori* configuration of MLE response selection solutions will most often only be as strong as the levels of uncertainty enveloping its stochastic constituents. In particular, it is important to concede that the modelling and optimisation techniques designed for incorporation in the DSS will only be useful and effective if the quality of information is able to meet a certain standard. Realistically speaking, the aim of this DSS should therefore rather be redefined as follows: *To provide an MLE response selection operator with real-time guidance on the macro-assignment of MLE resources with respect to the seemingly best course of action to be taken given the nature, quantity and quality of relevant information available at present.*

6.1.1 General sources of uncertainty

In order to better understand the inherent stochastic nature of the MLE response selection problem, it is first necessary to discuss the general sources of uncertainty that the MLE response selection DSS must accommodate. Three such sources particularly stand out. As is often the case with dynamic semi-automated DSSs in general, the first and most important uncertainty factor in the configuration and implementation of desired routing solutions deals with the quantity, format, accuracy and timeliness of input information received from the external data sources. As mentioned earlier, the performance of an MLE response selection DSS will only be as good as the standard of information provided to it, and so the general level of performance of external systems of a given coastal nation will therefore ultimately determine whether such a DSS is worth implementing.

The second source of uncertainty has to do with human error, in which various role players (such as decentralised operators or MLE resource operators) fail to implement the solution proposed by the DSS and MLE response selection operator in the recommended manner. A differentiation is, however, made between situations in which a human role player presumably unintentionally operates in ways that lead to a deterioration of the currently implemented solution, and in which a human role player deliberately takes actions in real time in order to adjust operations in certain ways so as to best conform to the current solution being implemented. Rational,

micro-managed course adjustments made by MLE resource operators as resources travel along their routes, for example, form part of such deliberate actions.

The remaining uncertainty factor is internal to the MLE response selection DSS and relates to the ability of the mathematical models to adequately represent and incorporate the various stochastic elements as part of the decision making process. Moreover, it may be argued that it is preferable to use as much external data sources of information as possible in order to render the models realistic. This is, however, only true if such information is filtered efficiently and applied correctly, and the DSS should be designed to make the best out of the information provided. Lastly, no matter how effective the other two uncertainty factors mentioned above may be managed in terms of minimising the magnitude of uncertainty, sudden and unpredictable events causing changes in input data are bound to occur during the course of the solution implementation, and it is therefore important that the MLE response selection DSS should be able to respond and adapt to these changes as effectively as possible.

6.1.2 Stochastic elements

In this section, the various non-deterministic elements of the stochastic MLE response selection problem are acknowledged and described in some detail. While most stochastic VRPs in the literature tend to focus on stochastic travel times or stochastic customer demands (see §2.6.4), the stochastic MLE response selection problem comprises, *inter alia*, a complex and rare case of stochastic VRP input data, in which customer locations are modelled as random variables in both time and space. Stochastic information pertaining to *customer visitation locations*, henceforth, refers to the uncertainty associated with the calculation of interception points at specific times in the future. As a result, it is easy to see that the lengths of routing arcs are also subject to uncertainty with respect to both distance and time. Travelling costs and response times are, consequently, also subject to a certain level of uncertainty. Moreover, the MLE response selection problem also includes stochastic customer demands, where there exists uncertainty in respect of the threat nature and threat intensity of VOIs. Similarly, stochastic *customer presence* in this context refers to the uncertainty in respect of the possibility that a VOI does not embody any threats at all (*i.e.* is a false alarm). Finally, it is acknowledged that stochastic information pertaining to *customer service time* (*i.e.* the time elapsed between the moment an MLE resource intercepts a VOI at sea and the moment it completes servicing it) is also present; these service times cannot be known with certainty *a priori*, because characteristics associated with VOIs, even those embodying the same threatening activities, may vary significantly from one VOI to another.

Moreover, it is acknowledged that several key stochastic elements are generated *via* sources that are external to the MLE response selection DSS and, therefore, over which the DSS has little or no control. These sources include stochastic information pertaining to VOI locations and VOI velocities (obtained from the MLE database *via* threat detection systems), VOI threat probabilities (obtained from the MLE database *via* threat evaluation systems) and MLE resources service times[1] (obtained from the MLE database *via* the MLE resource assignment external infrastructure system).

An interesting but unfortunate characteristic of the stochastic MLE response selection problem is that certain stochastic elements are themselves derived from other stochastic elements, which are themselves derived from further stochastic elements, and so on. These augmented stochastic elements are, consequently, subject to greater levels of uncertainty, as illustrated in Figure 6.1.

[1]This refers to the probability distribution of the time taken by a given MLE resource to service a VOI embodying a specific type of threat.

In this diagram, a node succeeded by one or more other nodes indicates that its underlying parameter estimators are (partially) derived from these preceding sources. In the remaining of this chapter, the MLE response selection stochastic elements are considered and analysed in greater detail.

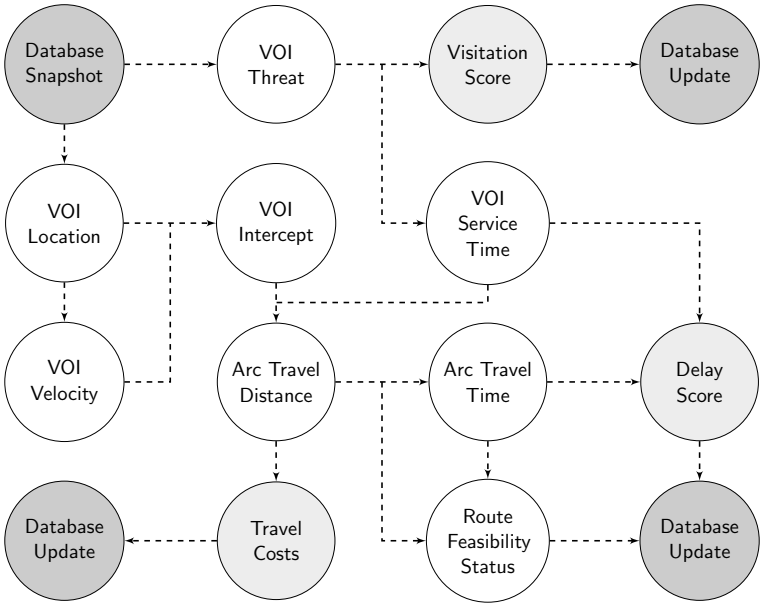


FIGURE 6.1: *Stochastic elements of the MLE response selection problem and the inter-computational connections between them.*

6.2 Stochastic information pertaining to VOI locations

Tracking and predicting the positions of VOIs in real time is probably the most crucial stochastic feature of the MLE response selection problem, particularly because it provides the basis from which most other MLE response selection stochastic elements are derived (see Figure 6.1). In this section, an effective manner in which to model the uncertainty pertaining to the positions of VOIs in space is pursued, after which the impact of threat detection systems performance on the tracking of VOI locations is discussed.

6.2.1 The bivariate Gaussian probability distribution

In the remainder of this chapter, and with limited loss of generality, it is assumed that threat detection systems are able to communicate the anticipated locations of VOIs in real time by employing the *bivariate Gaussian probability distribution* [163] — a very popular case of elliptical probability distributions for use in estimating the locations of objects in planar spaces [132, 137]. If the coordinates (X_i, Y_i) represent random variables modelling the position of VOI i along the axes of some two-dimensional global frame of reference, then $(X_i, Y_i) \sim \mathcal{N}(\mu_{iX}, \mu_{iY}, \sigma_{iX}^2, \sigma_{iY}^2, \rho_i)$ with joint probability density function

$$f_{\mathcal{N}}(x_i, y_i) = \frac{\exp\left[-\frac{1}{2(1-\rho_i^2)}\left(\frac{(x_i-\mu_{iX})^2}{\sigma_{iX}^2} + \frac{(y_i-\mu_{iY})^2}{\sigma_{iY}^2} - \frac{2\rho_i(x_i-\mu_{iX})(y_i-\mu_{iY})}{\sigma_{iX}\sigma_{iY}}\right)\right]}{2\pi\sigma_{iX}\sigma_{iY}(1-\rho_i^2)^{\frac{1}{2}}}.$$

For the sake of clarity, this probability distribution may, alternatively, be formulated as $(X_i, Y_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where

$$\boldsymbol{\mu}_i = \begin{bmatrix} \mu_{iX} \\ \mu_{iY} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_{iX}^2 & \rho_i \sigma_{iX} \sigma_{iY} \\ \rho_i \sigma_{iX} \sigma_{iY} & \sigma_{iY}^2 \end{bmatrix}.$$

In this formulation, the vector $\boldsymbol{\mu}_i$ contains the central tendencies of X_i and Y_i along each axis of the global coordinates system, while σ_{iX}^2 and σ_{iY}^2 represent the variation around the central tendencies of these random variables, respectively. These spreads of uncertainty are often referred to as the *Gaussian noise* of the distribution. Here, while the central tendencies of the VOI are analogous to its *expected location* in space, the spread factors σ_{iX} and σ_{iY}, or *standard deviations*, estimate the potential distance-measured error factors incurred along each axis around the expected location of the VOI. Furthermore, the parameter $\rho_i \in (-1, 1)$ reflects the degree of correlation between X_i and Y_i.

A very useful property of this joint probability distribution is that both random variables may be described, on a more fundamental level, by the set of linear equations

$$\begin{aligned} X_i &= \sigma_{iX} U_i + \mu_{iX}, \\ Y_i &= \sigma_{iY} \left(\rho_i U_i + V_i \sqrt{1 - \rho_i^2} \right) + \mu_{iY}, \end{aligned}$$

where $U_i \sim \mathcal{N}(0, 1)$ and $V_i \sim \mathcal{N}(0, 1)$ are independent random variables governed by standard univariate Gaussian distributions. It follows that both X_i and Y_i may individually be formulated as the Gaussian distribution $X_i \sim \mathcal{N}(\mu_{iX}, \sigma_{iX}^2)$ and $Y_i \sim \mathcal{N}(\mu_{iY}, \sigma_{iY}^2)$, and these distributions may be derived from their joint bivariate Gaussian distribution by deriving the *marginal probability density functions*

$$f_{\mathcal{N}X_i}(x_i) = \int_{-\infty}^{\infty} f_{\mathcal{N}}(x_i, y_i) d_{y_i}$$

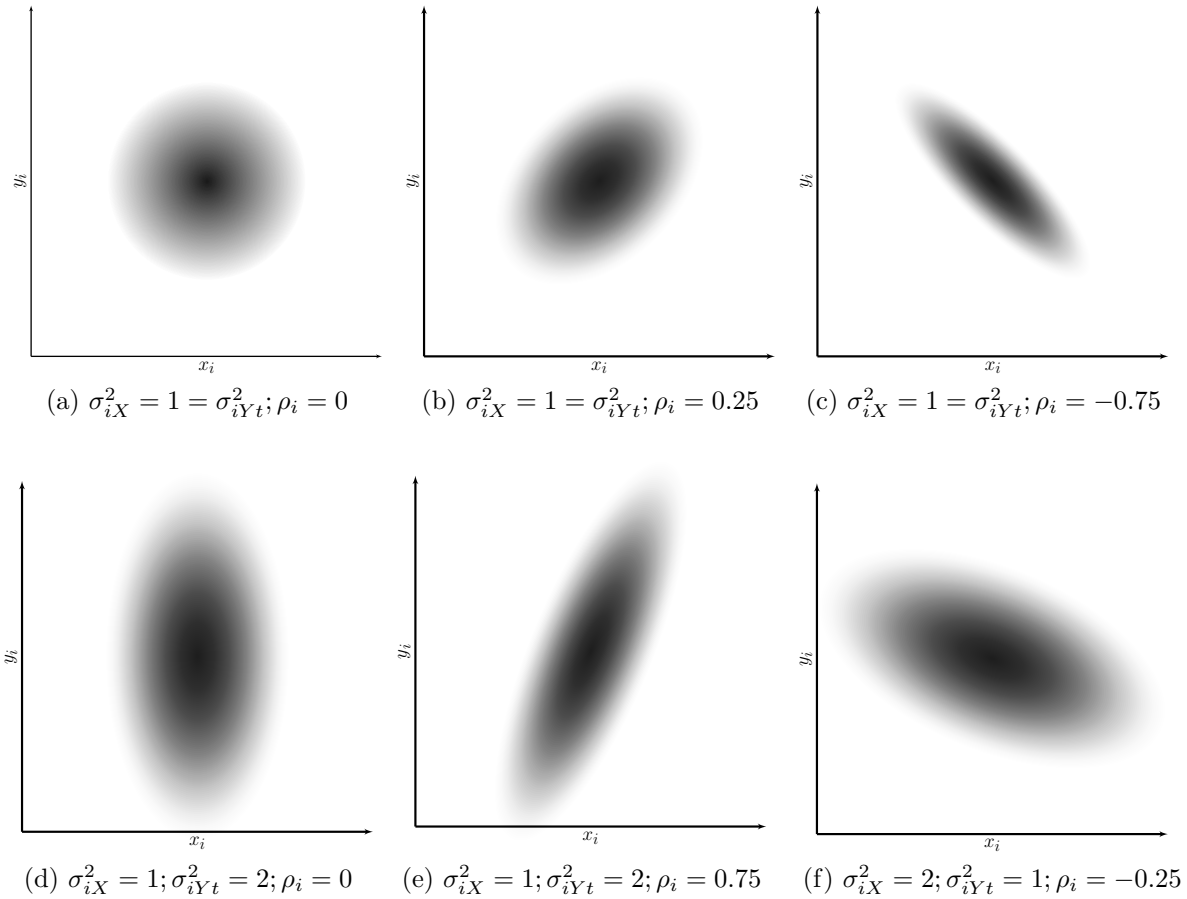
and

$$f_{\mathcal{N}Y_i}(y_i) = \int_{-\infty}^{\infty} f_{\mathcal{N}}(x_i, y_i) d_{x_i}.$$

Using these marginal probability distributions, the standard deviation of the variable X_i (and, similarly, of the random variable Y_i) may be expressed as

$$\sigma_{iX} = \sqrt{\sigma_{iX}^2} = \sqrt{\int_{(X_i, Y_i)} (x_i - \mu_{iX})^2 f_{\mathcal{N}X_i}(x_i) d_{x_i}}.$$

The contours of the bivariate Gaussian distribution may be represented visually as concentric ellipses growing in size and decreasing in probability around the locus defined by the central tendencies of the random variables. The boundaries of these ellipses are known as their *elliptic contours*. Additionally, the imaginary axis running along the longer diameter of an elliptic contour is known as its *major axis*, while its perpendicular bisector counterpart is known as the *minor axis*. The orientation and size of these elliptical distributions are purely defined by the nature of the covariance matrix $\boldsymbol{\Sigma}_i$. Illustrative examples of the impact of the covariance matrix on the nature of the bivariate Gaussian distribution are presented in Figure 6.2. Here, the probability that VOI i is located inside an infinitesimally small area on the two-dimensional continuum is quantified by the intensity of shading inside that area. It is interesting to note, for use later in the chapter, that all points located on a specific elliptic contour are shaded with the same intensity.



(a) $\sigma_{iX}^2 = 1 = \sigma_{iYt}^2; \rho_i = 0$ (b) $\sigma_{iX}^2 = 1 = \sigma_{iYt}^2; \rho_i = 0.25$ (c) $\sigma_{iX}^2 = 1 = \sigma_{iYt}^2; \rho_i = -0.75$

(d) $\sigma_{iX}^2 = 1; \sigma_{iYt}^2 = 2; \rho_i = 0$ (e) $\sigma_{iX}^2 = 1; \sigma_{iYt}^2 = 2; \rho_i = 0.75$ (f) $\sigma_{iX}^2 = 2; \sigma_{iYt}^2 = 1; \rho_i = -0.25$

FIGURE 6.2: *Bivariate Gaussian probability distribution illustrations with different covariance matrices.*

As ρ_i increases positively, the bivariate Gaussian distribution is stretched diagonally, forming elliptical isopleths with positively sloped major axes. Conversely, as ρ_i increases negatively, these isopleths are governed by negatively sloped major axes. Larger variance spreads stretch the elliptic contours along their major and minor axes, and the smaller the variance of a random variable is along its axis, the more "tightly peaked" the Gaussian bell in that dimension is. The case in which these two random variables are uncorrelated or independent (*i.e.* whenever $\rho_i = 0$) involves no apparent association between X_i and Y_i. Intuitively, this is equivalent to stating that no information is to be gained in investigating the "distance" from X_i to μ_{iX} by knowing the value of Y_i, and *vice versa* [163]. On the other hand, a strong correlation between the two random variables suggests that the average "distance" of the pair (X_i, Y_i) in the joint probability distribution to the locus of central tendencies is expected to be rendered smaller by this association. In other words, knowledge of one diminishes uncertainty in the other one for cases in which these random variables are highly correlated.

Another benefit of adopting this probability distribution is that the isomorphism mapping points from a real-life global coordinate system (*e.g.* a geodetic or polar coordinate system) to a more simplified scale used for computing these probability distributions is both relatively simple and one-to-one. For example, a coastal nation may wish to employ an isomorphism that converts some real-life maritime coordinate system to a temporary, normalised, shifted local Cartesian space for the purpose of efficiently computing response selection interception trajectories.

A significant drawback of adopting a bivariate Gaussian distribution to model the location of

objects in an MLE context, however, occurs in cases where VOIs are not cruising on the open sea, but are rather moving close to land or other infeasible locations. In such cases, subsets of the locations surrounding the expected location of a VOI will not be assigned zero probability weights, although there really is a zero chance (with a few exceptions, such as amphibious vehicles) that a VOI location will, at any point, be observed in such a subset. These cases therefore require the implementation of alternative location modelling features.

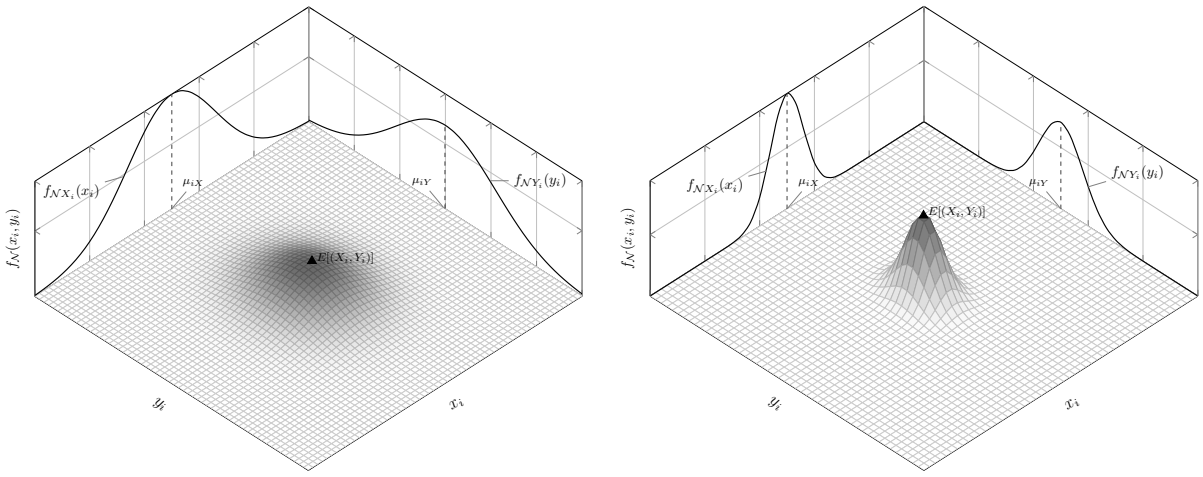
6.2.2 On threat detection performance and VOI location uncertainty

As discussed in §3.1.1, threat detection systems are expected to transmit certain input data to the MLE database in a specific format, and in a timely manner, via the attribute management system. VOI coordinates at sea, VOI headings, VOI speeds and VOI predicted travel paths are examples of such data. This information may then used as part of MLE response selection operations in order, *inter alia*, to establish the feasible domain of a problem instance, detect disturbances, and make necessary solution progression micro-adjustments. The reader is referred to the work of du Toit [49] to gain some insight into the process of deriving threat detection information related to location, movement and trajectory paths of VOIs.

This information transfer process may, however, significantly vary from one coastal nation to another as a result of the different natures of MLE threat detection systems in place. Indeed, as mentioned earlier, the ability of MLE response selection systems to receive adequate VOI information in real time is significantly constrained by the ability of threat detection systems to derive such information.

In order to achieve effective MLE response selection operations, real time VOI locations ought to be pinpointed in space as accurately as possible with as little margin for error as possible, and this is also critical when anticipating future VOI locations. The quality and level of resolution of radars as well as their general distance from VOIs are examples of factors that may have an impact on the level of uncertainty pertaining to VOI locations in real time. Examples of "weak" and "competent" threat detection systems implemented toward modelling the probability distribution of the location of the same VOI at the same time is illustrated in Figure 6.3. Note that this VOI location follows a bivariate Gaussian probability distribution. Here, it is easy to see that the two univariate Gaussian bells are more stretched out in Figure 6.3 (a) as a result of higher uncertainty around the corresponding marginal probability density function mean values, μ_{iX} and μ_{iY}, which is an indication of lesser adequate threat detection systems.

Fortunately, there exist many effective techniques to consolidate the data obtained from observations made by various (independent) sources in order to predict the position of an object at sea more accurately. This is particularly the case whenever a coastal nation has multiple radar sources at its disposal that do not typically perform very well on their own (as illustrated in Figure 6.3 (a)). The reader is referred to the work of Stroupe *et al.* [137] to gain insight into how this is achieved. Here, object positions collected by multiple individually operated robots are modelled as bivariate Gaussian distributions. A consolidation technique employed, called the *multi-distribution merging process*, then merges these probability distributions as a means to achieve greater effective sensor coverage of the environment, in turn improving the accuracy of object position estimates. This technique is analogous to using multiple threat detection sources as a way of assessing the estimated position of VOIs in real time. That is, information collected from multiple points of view can reduce uncertainty, improve accuracy and tighten confidence intervals. Another example of such a technique is the cooperative merging of several hundreds of super-telescopes around Earth in order to improve the accuracy of images gathered in deep outer space [155].



(a) *Weak MLE threat detection systems result in higher uncertainty with respect to the estimated locations of VOIs in general.*

(b) *Competent MLE threat detection systems result in higher accuracy with respect to the estimated locations of VOIs in general.*

FIGURE 6.3: *An illustration of two different threat detection systems used to model the probability distribution of the location of (the same) VOI i along a two-dimensional coordinate system at a certain point in time. Here, the joint probability density function $f_{\mathcal{N}}(x_i, y_i)$ follows a bivariate Gaussian distribution with corresponding univariate marginal probability density functions $f_{\mathcal{N}X_i}(x_i)$ and $f_{\mathcal{N}Y_i}(y_i)$ projected onto their respective planes.*

6.3 Stochastic information pertaining to visitation locations

Although the information pertaining to VOI location tracking in real time discussed in the previous section is important for use in MLE response selection operations, it is really only important for trajectory adjustments on a micro-management level as well as for the detection of sudden, unanticipated changes in VOI itineraries for use in problem instance reinitialisation processes. For the purpose of configuring visitation routes, however, it is further required to anticipate the locations of VOIs in the future, and this is unarguably of utmost importance. Indeed, as described in Chapter 4, these anticipated locations are used in the mathematical modelling process to compute visitation locations — points in space and time on the visitation routes of active MLE resources where interception with the scheduled VOIs take place.

6.3.1 Anticipated VOI trajectories

Intuitively speaking, anticipated VOI locations may be thought of as 2-dimensional T-tuples corresponding to the projections of the anticipated central tendencies of VOI i at specific time instants, modelled as a time series $\{(X_i, Y_i)_t,\ t = 1, 2, \ldots, T\}$, for all $i \in \mathcal{V}^e$. This set is defined here as an *anticipated VOI trajectory* — a time-ordered set of states of *anticipated VOI locations* at various points in the future which are governed by bivariate Gaussian probability distributions. Although the long term accuracy of predicting anticipated VOI locations is expected to be relatively poor, it is nevertheless important that the pre-configured length of this time series (*i.e.* the parameter T) be large enough to allow for MLE response selection operations involving the estimation of interception points along MLE resource visitation routes. Trajectory micro-adjustments may nevertheless be carried out accordingly by MLE resource operators in time as the routes progress and the actual anticipated interception points with VOIs scheduled for

visitation become more pronounced (as these become progressively nearer in the future than when they were assessed at the beginning of the time stage).

Anticipated VOI trajectories may typically be derived by various techniques that combine an appropriate amount of relevant information for best predictability. Such information may include, for example, the estimated size, current location and current velocity of the VOI, together with its historical trajectory and speculations in respect of its possible activities and intentions at sea. As mentioned in §6.2, however, the level of accuracy of anticipated VOI trajectories will, in general, greatly depend on the accuracy of threat detection systems and expertise of operators to perform these tasks. This problem is often complicated in real life as the expected future location of a VOI may typically only be known indirectly through a sequence of relative frames of reference with increasing uncertainty [6].

Examples of techniques employed in the literature to derive anticipated VOI trajectories include *multi-modal Gaussian mixture models* [6] and *hidden Markov models* [49]. An in-depth analysis of anticipated VOI trajectories is outside the scope of this dissertation. Based on what was uncovered in the literature, it is, nevertheless, necessary to acknowledge that this undertaking is by no means a trivial task. It is therefore assumed in the remainder of this chapter that threat detection systems are able to derive this information to a certain degree of predictability and make it available to the MLE response selection DSS in real time.

6.3.2 On the representation of anticipated VOI velocity vectors

Similarly to predicting VOI locations in real time, deriving VOI anticipated velocities should ideally be carried out by combining various techniques as a means to achieve maximum efficiency. Such techniques include *polynomial* and *spline interpolation* [161], *compound approximate transformations* [132], *Bayesian time series analysis* (for the evaluation of posterior probability distributions) [121] and *dynamic time warping* [49]. An in-depth analysis of these a processes is, once again, beyond the scope of this dissertation. The focus in this section is rather on the presentation of various useful formats in which MLE threat detection systems may model anticipated VOI velocity vectors for use in MLE response selection operations.

Because these velocity vectors are not characterised with explicit sets of parameters (except for the chronological progression of the object at points on a time line), their derivation resides in the modelling realm of *non-parametric* modelling [132]. Perhaps the most appropriate mathematical modelling approach towards determining the chronological progression of such objects in both space and time involves the derivation of pairs of parametric expressions in terms of a time parameter, such that each of these expressions represents the anticipated progression of a VOI as a function of time with respect to one of two global axes. This is a useful representation, as these expressions will simultaneously also express the anticipated locations of VOIs over time.

Three general styles of anticipated VOI velocity vector representations are presented in the remainder of this section. If the anticipated rate of change of displacement of a VOI over a pre-defined period of time is fixed, then its velocity vector representation is referred to as *static*. On the other hand, a velocity vector representation is said to be *dynamic* if the VOI is expected to change speed or direction at any point in the near-future. The choice of velocity vector representation will typically vary based on subjective preference of operators, quantity of historical locations data gathered, operators expertise and general prediction capabilities and accuracy of the threat detection system models.

Static representation of velocity vectors

Two methods of static anticipated VOI velocity vectors are presented here. The first involves treating the rate of change of displacements along each of two global axes independently, that is defining VOI i's anticipated velocity vector $\tilde{\boldsymbol{v}}_i = [\varsigma_{ix}\ \varsigma_{iy}]$, where ς_{ix} is the anticipated (fixed) speed component along the x-axis and ς_{iy} is the anticipated speed component along the y-axis. The anticipated location of this VOI as a function of time with respect to the global frame of reference is given by

$$(x_i(t), y_i(t)) = (x_i(0) + t\varsigma_{ix}, y_i(0) + t\varsigma_{iy}).$$

The second method of static velocity vector representation involves assessing the estimated heading θ_i of VOI i with respect to a global x-axis together with its estimated speed ς_i relative to the plane in order to form the vector $\tilde{\boldsymbol{v}}_i = [\theta_i\ s_i]$. This representation is particularly useful from a visual point of view on operator screens, where a vector arrow may point out of the VOI such that the direction of this arrow is governed by θ_i and such that its length is directly proportional to ς_i. In this representation, the anticipated location of the VOI as a function of time with respect to the global frame of reference is given by

$$(x_i(t), y_i(t)) = (x_i(0) + t\varsigma_i \cos\theta_i, y_i(0) + t\varsigma_i \sin\theta_i).$$

Although static velocity vector representations are relatively easy to configure, they are most likely not a very accurate representation of reality[2], particularly if the time period over which estimates of the velocity of the VOI is performed is long. This representation nevertheless works best when few data points are available and linear interpolation is the best available option to predict the movement of VOIs.

Dynamic representation of velocity vectors

Alternatively, it is advocated that parametric representations of the anticipated velocity vectors of VOIs be configured more generically by means of higher-degree polynomials (*i.e.* preferably such that at least one of the two vector components be at least quadratic). For example, let the anticipated location of VOI i as a function of time be given by

$$(x_i(t), y_i(t)) = (a_{i0} + a_{i1}t + a_{i2}t^2 + \ldots + a_{iN_{ix}}t^{N_{ix}}, b_{i0} + b_{i1}t + b_{i2}t^2 + \ldots + b_{iN_{iy}}t^{N_{iy}}),$$

where $a_{i0}, \ldots, a_{iN_x}$ and $b_{i0}, \ldots, b_{iN_y}$ are real coefficients. In the case where $N_{ix} = 1 = N_{iy}$, the representation conforms to the formulation of the first type of static velocity vector representation discussed above, where $a_{i1} = s_{ix}$ and $b_{i1} = s_{iy}$. Of course, the case where $N_{ix} = 0 = N_{iy}$ suggests the presence of an immobile VOI.

Given a set of parametric expressions in such a format, its corresponding anticipated VOI velocity vector as a function of time is given by

$$\tilde{\boldsymbol{v}}_i(t) = \left[\frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\ \ \frac{\mathrm{d}y_i(t)}{\mathrm{d}t}\right].$$

Moreover, one may also wish to derive the anticipated speed and acceleration vector of the VOI as a function of time for use in MLE response selection operations. Its speed component at any

[2]The level of accuracy of the VOI velocity prediction models can be measured retrospectively by comparing the *a priori* anticipated VOI trajectory with its *a posteriori* trajectory path using appropriate metrics such as measuring the sum of the areas enclosed between the two curves over a pre-defined finite period of time.

given time is then easily computed as

$$||\tilde{\boldsymbol{v}}_i(t)|| = \sqrt{\left(\frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}y_i(t)}{\mathrm{d}t}\right)^2},$$

while its bearing component is derived using the *arc tangent* function

$$\tilde{\theta}_i(t) = \arctan\left(\frac{\mathrm{d}y_i(t)}{\mathrm{d}t}\left(\frac{\mathrm{d}x_i(t)}{\mathrm{d}t}\right)^{-1}\right) \in [-\pi, \pi]$$

and its acceleration vector is simply

$$\tilde{\boldsymbol{a}}_i(t) = \left[\frac{\mathrm{d}^2x_i(t)}{\mathrm{d}t^2} \;\; \frac{\mathrm{d}^2y_i(t)}{\mathrm{d}t^2}\right].$$

Step function representation

To avoid the difficulties that may accompany the evaluation of higher-degree polynomials, as well as possibly improving prediction accuracy, an anticipated VOI velocity vector may, alternatively, be broken down into a step function consisting of a time sequence of pre-defined lower-degree polynomials. This is particularly useful in cases where the VOI is strongly expected to experience sudden significant changes in direction and/or velocity at specific points in the future. Such predictions may, for example, be the result of certain geographical or environmental features crossing paths with the anticipated trajectory of the VOI, or of the known threatening activities and intentions of the VOI that give away clear indications as to where it is expected to make sudden dramatic changes in velocity.

Such an anticipated VOI velocity representation may be expressed as a continuous sequence of Ξ consecutive anticipated VOI velocity vectors defined over specific time intervals $[0, t_1], [t_1, t_2], \ldots, [t_{T-1}, \Xi]$, that is

$$\tilde{\boldsymbol{v}}_i(t) = \begin{cases} \tilde{\boldsymbol{v}}_i^1(t)\,, & \text{if } 0 \leq t \leq t_1, \\ \tilde{\boldsymbol{v}}_i^2(t)\,, & \text{if } t_1 \leq t \leq t_2, \\ \quad\vdots & \quad\vdots \\ \tilde{\boldsymbol{v}}_i^{\Xi}(t)\,, & \text{if } t_{\Xi-1} \leq t_{\Xi}. \end{cases}$$

6.3.3 Uncertainty factors pertaining to anticipated VOI locations

In this section, a discussion is conducted on the major factors impacting the estimated level of uncertainty of anticipated VOI locations. Overall, a trade-off has to be found between the extent to which these errors may be accounted for and the rate at which new problem instances are triggered (*i.e.* time stage termination) due to unforeseen input data deviations caused by these factors. In particular, it is required to assess the potential extent of influence of such changes on the quality of a currently implemented solution. It is expected that the following six factors may influence the intensity of noise whilst in the process of configuring the probability distributions corresponding to the set of anticipated VOI locations $\{(X_i, Y_i)_t\}$:

1. The performance level of threat detection external infrastructure,
2. The performance level of threat detection systems in the derivation of anticipated VOI velocity vectors,

3. The stretch of prediction time into the future,
4. The environmental conditions (at present and anticipated) and geographical layout of the coastal nation's territorial waters,
5. The actual nature of the accumulated historical information gathered on the VOI, and
6. The actual, calculated anticipated velocity vector of the VOI.

In the above list, Factors 1–4 are rather straightforward. The underlying level of noise in these cases is generally induced independently of the nature of the VOI. Factors 5 and 6, on the other hand, are more intricate. Here, the underlying level of noise is determined based on certain characteristics that are rather internal to the VOI itself.

Factor 1 follows directly from the discussion in §6.2.2. More specifically, because VOI historical positions detected by radar systems are employed in the calculation of anticipated VOI velocities (and hence anticipated VOI locations), the noise detected in these historical locations in space will have a direct impact on the amount of noise surrounding the anticipated locations of the VOI. In addition, the rate at which radar are able to transmit information on an object at sea will influence the quantity of historical data obtained, in turn influencing the quality of future predictions on the whereabouts of the object. Polynomial interpolation is an example of a technique used in predicting the future location of an object which significantly benefits from a larger quantity of historical data points.

Similarly, Factor 2 refers to the performance level of the internal automated and/or semi-automated threat detection systems in place during the process of deriving anticipated VOI velocity vectors in general, as well as the expertise of threat detection operators responsible for assisting with these tasks. Like numerous other similar systems of this kind, it is suggested that a steep learning curve may exist whilst performing these tasks over time. In particular, threat detection systems should compare actual *a posteriori* information to *a priori* predictions. A continual system improvement mechanism, based on this comparison, should take place so that predictions may become more accurate in the future. Furthermore, threat detection systems should continually communicate observations to threat evaluation systems. Threat evaluation operators may, for example, be interested in studying the probability that a VOI embodies a certain type of threat provided that it has moved and is expected to move in a certain, anticipated way. Such analyses may ultimately improve threat evaluation predictions over time.

Factor 3 relates to the notion that noise tends to grow monotonically as predictions are made further into the future. That is, all other factors aside, the uncertainty surrounding anticipated VOI locations with respect to the global frame of reference is expected to grow as a function of time. Interestingly, the objective function responsible for minimising the weighted response selection time of the assignment of MLE resources onto a subset of VOIs (see Objective II in §4.2.3) may analogously be employed to minimise the repercussions caused by the implications of this factor. This is particularly the case whenever the duration of a time stage is long.

Factor 4 takes into account these unpredictable or hard to account for environmental phenomena that may force a VOI to alter its course at some point in the near future. Moreover, various geographical layouts may require the VOI to enter, avoid or navigate through certain bounded maritime spatial zones, such as the presence of small islands or shallow waters.

Factor 5 is related to the impact of the actual nature of the data pertaining to the historical information of the VOI on its predictability level in the future. It is suggested that a VOI that has behaved unpredictably (to some extent) in the past is also more likely to replicate such behaviour in the future. This may include, for example, VOIs having exhibited frequent

and/or significant unpredicted rates of change of velocity along their historical tracks, or high uncertainty with respect to the type of threat(s) that some VOIs are speculated to embody, in turn causing its future whereabouts to be more speculative as well.

Finally, Factor 6 relates to the actual nature of the anticipated velocity vector assigned to a VOI and, in particular, to its anticipated speed component. More specifically, slight unanticipated bearing deviations from the anticipated bearing vector component ($\tilde{\theta}_i(t)$) and, consequently, from the anticipated VOI locations, are expected to have a larger impact on the margin of error in anticipated VOI locations over a specific time frame whenever VOIs navigate at relatively high speeds. As a result, there may exist a positive correlation between the anticipated speed of a VOI and its anticipated location uncertainty in the future.

6.3.4 Derivation of anticipated VOI location probability distributions

For the purpose of enforcing the abstract link between MLE threat detection and MLE response selection systems, it is important to step slightly outside the scope of this dissertation and elaborate briefly on the process according to which threat detection systems may be able to derive the probability distributions associated with the anticipated locations of VOIs. While the central tendencies of these distributions may be derived with relative ease from the anticipated VOI velocity vectors (or, analogously, from the anticipated VOI location parametric representations, as discussed in §6.3.2), deriving the parameter estimators representing the degree of uncertainty around these central tendencies forms another thick layer of complexity to the already highly-stochastic nature of the MLE response selection problem. The focus in the remainder of this section will therefore evolve around determining such estimators in a format that is both realistic and adequate to the formulation of stochastic MLE response selection model components.

In §6.2.1, it was suggested that the stochastic representation of the real time locations of VOIs may be modelled effectively according to a bivariate Gaussian probability distribution. Following this suggestion, consider the pair of random variables $(X_i, Y_i)_t \sim \mathcal{N}(\boldsymbol{\mu}_{it}, \boldsymbol{\Sigma}_{it})$ with probability density function $f_{\mathcal{N}}(x_i, y_i)_t$, for all $t \in \{1, 2, \ldots, T\}$. It is important to note that the consolidation $\boldsymbol{\mu}_{it} = E[(X_i, Y_i)_t] = (x_i(t), y_i(t))$ between the anticipated VOI locations and its future (Gaussian) central tendencies follows as a result of this construction. Moreover, from Factor 3 in §6.3.3, it is agreed that the condition $\sigma_{ix(t+1)} > \sigma_{ixt}$ and $\sigma_{iy(t+1)} > \sigma_{iyt}$ must hold for all $t \in \{0, \ldots, T-1\}$ in general, where σ_{ixt} and σ_{iyt} are the anticipated standard deviations associated with the random variables $(X_i, Y_i)_t$.

The derivation of the associated distribution covariance matrix is conducted next. Perhaps the most effective way in which threat detection systems are able to model the uncertainty surrounding the location of a VOI at some point in the future according to a bivariate Gaussian probability distribution is to orientate the major axes of its underlying elliptic contours in alignment with the anticipated velocity of the VOI a that time. It follows that the bearing of the VOI will always be aligned to the major axis of its Gaussian distribution at that time in such a configuration.

Moreover, it may be convenient to model the anticipated central tendencies and Gaussian noise of the VOI in respect of a strategic two-dimensional temporal local set of coordinate axes, or *local frame* of reference. To this end, it is proposed that the local frame of reference be modelled with the ordered basis $\{\breve{x}_{it^*}, \breve{y}_{it^*}\}$ configured in such a way as to have the local x-axis and y-axis parallel and perpendicular to the bearing of the anticipated VOI velocity at time $t^* \geq 0$, respectively. A visual representation of this construction is illustrated in Figure 6.4. This particular choice of local axes has the benefit of avoiding any correlation between the two random variables $\breve{X}_{it^*}$ and $\breve{Y}_{it^*}$, which is a difficult aspect of the distribution covariance

that threat detection systems therefore need not concern themselves with. This local bivariate Gaussian distribution may therefore be modelled as $(\breve{X}_{it^*}, \breve{Y}_{it^*}) \sim \mathcal{N}(\breve{\boldsymbol{\mu}}_{it^*}, \breve{\boldsymbol{\Sigma}})$, where $\breve{\boldsymbol{\mu}}_{it^*} = \boldsymbol{\mu}_{it^*}$ and

$$\breve{\boldsymbol{\Sigma}}_{it^*} = \begin{bmatrix} \breve{\sigma}^2_{iXt^*} & 0 \\ 0 & \breve{\sigma}^2_{iYt^*} \end{bmatrix}. \tag{6.1}$$

In (6.1), the values $\breve{\sigma}^2_{iXt^*}$ and $\breve{\sigma}^2_{iXt^*}$ represent the estimated error margins along the $\breve{x}_{it*}$ and $\breve{y}_{it*}$ axes (the major and minor axes of this distribution), respectively.

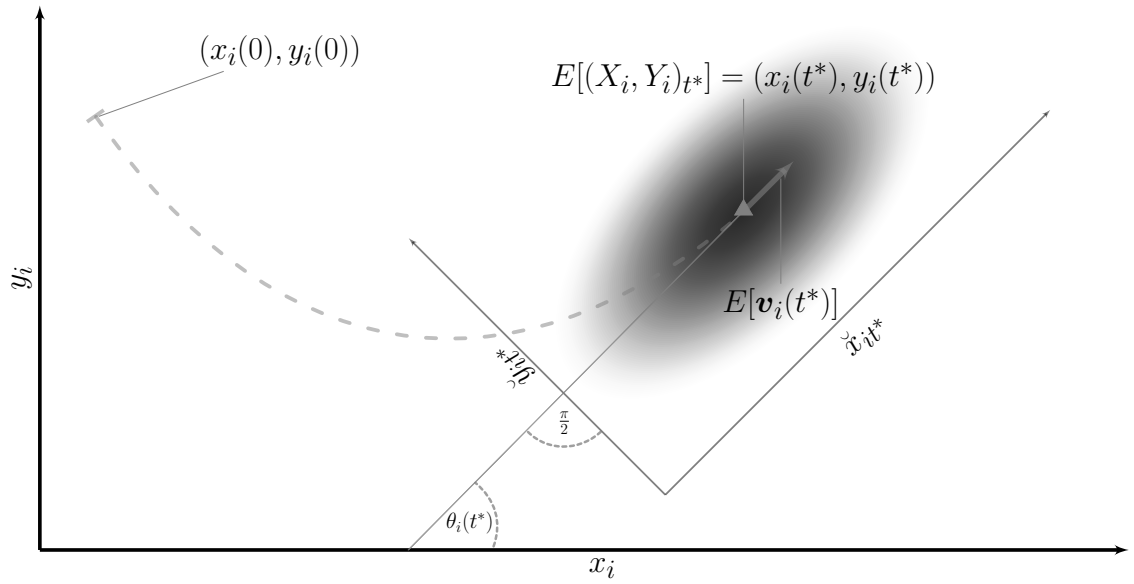


FIGURE 6.4: *A bivariate Gaussian probability distribution relative to a local set of coordinates.*

For use in MLE response selection operations, however, it is necessary to transform this probability distribution representation of anticipated VOI locations to the global frame of reference, referred to here as its *canonical form*. In their research, Smith *et al.* [132] defined an *uncertainty transformation* as an estimated mean relation of one local coordinate frame relative to some defining global coordinate frame together with a covariance matrix that expresses the uncertainty of such an estimate.

Transforming the probability distribution from this local frame to its canonical form may be achieved *via* a linear isomorphism procedure. It is easy to see that such a transformation may be achieved by performing a rotation of $-\theta_{it^*}$ radians of all points around the central tendencies of the distribution, which corresponds to an angular rotation of the covariance matrix in which the parameter ρ_{it} allows for the major and minor axes of a Gaussian ellipse to be rotated. This rotation isomorphism may be achieved by means of the two-by-two matrix

$$\mathbf{W} = \begin{bmatrix} \cos\theta_{it^*} & \sin\theta_{it^*} \\ -\sin\theta_{it^*} & \cos\theta_{it^*} \end{bmatrix}$$

with transpose

$$\mathbf{W}^T = \begin{bmatrix} \cos\theta_{it^*} & -\sin\theta_{it^*} \\ \sin\theta_{it^*} & \cos\theta_{it^*} \end{bmatrix}.$$

By computing the Jacobian matrix of the transformation equations and employing the properties of eigenvectors, the identity

$$[x_i\ y_i]\,\boldsymbol{\Sigma}^{-1}\begin{bmatrix} x_i \\ y_i \end{bmatrix} = [x_i\ y_i]\,\mathbf{W}^T\breve{\boldsymbol{\Sigma}}^{-1}\mathbf{W}\begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

holds true for all transformations of this type [107, 132, 137], where (x_i, y_i) is a vector of random variables governed by a bivariate Gaussian probability distribution[3]. It is a well-known fact that

$$\mathbf{W}^T = \mathbf{W}^{-1}$$

[132]. The transformation of the local covariance matrix to its canonical form is therefore given by

$$\begin{aligned}
\boldsymbol{\Sigma} &= \left(\mathbf{W}^T \breve{\boldsymbol{\Sigma}}^{-1} \mathbf{W}\right)^{-1} \\
&= \left(\breve{\boldsymbol{\Sigma}}^{-1} \mathbf{W}\right)^{-1} (\mathbf{W}^T)^{-1} \\
&= \mathbf{W}^{-1} \left(\breve{\boldsymbol{\Sigma}}^{-1}\right)^{-1} (\mathbf{W}^T)^{-1} \\
&= \mathbf{W}^{-1} \breve{\boldsymbol{\Sigma}} (\mathbf{W}^T)^{-1} \\
&= \mathbf{W}^T \breve{\boldsymbol{\Sigma}} (\mathbf{W}^{-1})^{-1} \\
&= \mathbf{W}^T \breve{\boldsymbol{\Sigma}} \mathbf{W}.
\end{aligned}$$

It follows that

$$\begin{aligned}
\boldsymbol{\Sigma} &= \begin{bmatrix} -\cos\theta_{it^*} & \sin\theta_{it^*} \\ -\sin\theta_{it^*} & -\cos\theta_{it^*} \end{bmatrix} \begin{bmatrix} \breve{\sigma}^2_{iXt^*} & 0 \\ 0 & \breve{\sigma}^2_{iYt^*} \end{bmatrix} \begin{bmatrix} -\cos\theta_{it^*} & -\sin\theta_{it^*} \\ \sin\theta_{it^*} & -\cos\theta_{it^*} \end{bmatrix} \\
&= \begin{bmatrix} -\breve{\sigma}^2_{iXt^*}\cos\theta_{it^*} & \breve{\sigma}^2_{iYt^*}\sin\theta_{it^*} \\ -\breve{\sigma}^2_{iXt^*}\sin\theta_{it^*} & -\breve{\sigma}^2_{iYt^*}\cos\theta_{it^*} \end{bmatrix} \begin{bmatrix} -\cos\theta_{it^*} & -\sin\theta_{it^*} \\ \sin\theta_{it^*} & -\cos\theta_{it^*} \end{bmatrix} \\
&= \begin{bmatrix} \breve{\sigma}^2_{iXt^*}\cos^2\theta_{it^*} + \breve{\sigma}^2_{iYt^*}\sin^2\theta_{it^*} & (\breve{\sigma}^2_{iXt^*} - \breve{\sigma}^2_{iYt^*})\sin\theta_{it^*}\cos\theta_{it^*} \\ (\breve{\sigma}^2_{iXt^*} - \breve{\sigma}^2_{iYt^*})\sin\theta_{it^*}\cos\theta_{it^*} & \breve{\sigma}^2_{iXt^*}\sin^2\theta_{it^*} + \breve{\sigma}^2_{iYt^*}\cos^2\theta_{it^*} \end{bmatrix}.
\end{aligned}$$

The variances of this distribution in respect of the global frame of reference are therefore given by

$$\sigma^2_{iXt^*} = \boldsymbol{\Sigma}_{11} = \breve{\sigma}^2_{iXt^*}\cos^2\theta_{it^*} + \breve{\sigma}^2_{iYt^*}\sin^2\theta_{it^*}, \tag{6.2}$$

$$\sigma^2_{iYt^*} = \boldsymbol{\Sigma}_{22} = \breve{\sigma}^2_{iXt^*}\sin^2\theta_{it^*} + \breve{\sigma}^2_{iYt^*}\cos^2\theta_{it^*}. \tag{6.3}$$

Moreover, the correlation between the random variables X_{it^*} and Y_{it^*} is given by the coefficient

$$\begin{aligned}
\rho_{it^*} &= \frac{\boldsymbol{\Sigma}_{12}}{\sigma_{iXt^*}\sigma_{iYt^*}} \\
&= \frac{\boldsymbol{\Sigma}_{21}}{\sigma_{iXt^*}\sigma_{iYt^*}} \\
&= \frac{(\breve{\sigma}^2_{iXt^*} - \breve{\sigma}^2_{iYt^*})\sin\theta_{it^*}\cos\theta_{it^*}}{\sqrt{\left(\breve{\sigma}^2_{iXt^*}\cos^2\theta_{it^*} + \breve{\sigma}^2_{iYt^*}\sin^2\theta_{it^*}\right)\left(\breve{\sigma}^2_{iXt^*}\sin^2\theta_{it^*} + \breve{\sigma}^2_{iYt^*}\cos^2\theta_{it^*}\right)}}.
\end{aligned} \tag{6.4}$$

6.3.5 A worked example

In order to validate the modelling approach described above for mapping the Gaussian noise around the central tendencies of anticipated VOI locations, a numerical example representing a hypothetical scenario is presented in this section.

[3]Interestingly, a higher proportion of similar research endeavors focus on the converse of this problem, in which the probability distribution of an object relative to a global frame of reference is required to be converted into a local one in order to obtain a diagonal covariance matrix.

Consider the i^{th} VOI whose anticipated trajectory has to be mapped in time and space over a three-hour period at regular half-hour intervals. Suppose that, based on historical information associated with the VOI, threat detection systems are able to estimate its anticipated velocity vector as

$$\tilde{\boldsymbol{v}}_i(t) = (5.4;\ -1.04t)\,,$$

where each vector component is measured in nautical miles per hour or *knots*. If the estimated position of the VOI with respect to the global frame of reference at time $t = 0$ is $(535, 104.7)$, the anticipated location of this VOI as a function of time is then given by

$$\tilde{\boldsymbol{\ell}}_i(t) = \left(535 + 5.4t;\ 104.7 - 0.52t^2\right),$$

its anticipated speed as a function of time is given by

$$||\tilde{\boldsymbol{v}}_i(t)|| = \sqrt{29.16 + 1.08t^2},$$

and its anticipated acceleration vector by

$$\tilde{\boldsymbol{a}}_i(t) = (0, -1.04).$$

Although the acceleration is constant, it does not mean the VOI is expected to accelerate constantly for three hours in a row. The parametric equation along the y-axis is nevertheless quadratic, thus merely suggesting that the VOI's rate of change of displacement along this axis is not expected to be constant during the course of the three hour period, but rather "ever so slightly" increasing over time. These data are tabulated in Table 6.1.

Time series	$\tilde{\boldsymbol{\ell}}_i(t)$	$\tilde{\boldsymbol{v}}_i(t)$	$\|\|\tilde{\boldsymbol{v}}_i(t)\|\|$	$\tilde{\theta}_i(t)$	$\tilde{\boldsymbol{a}}_i(t)$
$(X_i, Y_i)_0$	$(535, 104.70)$	$(5.4; 0.00)$	5.40	0	$(0, -1.04)$
$(X_i, Y_i)_{0.5}$	$(537.7, 104.57)$	$(5.4; -0.52)$	5.42	-5.49	$(0, -1.04)$
$(X_i, Y_i)_1$	$(540.4, 104.18)$	$(5.4; -1.04)$	5.50	-10.90	$(0, -1.04)$
$(X_i, Y_i)_{1.5}$	$(543.1, 103.53)$	$(5.4; -1.54)$	5.62	-16.11	$(0, -1.04)$
$(X_i, Y_i)_2$	$(545.8, 102.62)$	$(5.4; -2.08)$	5.79	-21.07	$(0, -1.04)$
$(X_i, Y_i)_{2.5}$	$(458.5, 101.45)$	$(5.4; -2.60)$	6.00	-25.71	$(0, -1.04)$
$(X_i, Y_i)_3$	$(551.2, 100.02)$	$(5.4; -3.12)$	6.24	-30.02	$(0, -1.04)$

TABLE 6.1: *Anticipated VOI characteristics in the form of a time series* $\{(X_i, Y_i)_t\}$.

Suppose that these anticipated VOI locations are each modelled as bivariate Gaussian probability distributions with respect to some temporary local sets of axes, constructed as described in §6.3.4. In addition, suppose that the threat detection systems are able to predict the Gaussian noise with respect to the local frame of references (namely the parameters $\breve{\sigma}^2_{iXt^*}$ and $\breve{\sigma}^2_{iYt^*}$) around the anticipated locations in the time series $\{(X_i, Y_i)_t\}$ as shown in Table 6.2. The results of applying (6.2)–(6.4) are also shown in Table 6.2. These results were computed in the software package MATLAB [147]. A visualisation of the resulting time series of anticipated VOI locations is provided in Figure 6.5.

6.3.6 The dynamic bivariate Gaussian probability distribution

While an effective procedure for modelling the anticipated locations of a VOI defined by the elements of the discrete set $\{(X_i, Y_i)_t\}$ was proposed in the previous section, it is duly acknowledged that, given the nature of the routing models of Chapter 4, VOI interception trajectories

Time series	μ_{iX}	μ_{iY}	$\breve{\sigma}^2_{iXt}$	$\breve{\sigma}^2_{iYt}$	$\breve{\rho}_i$	σ^2_{iXt}	σ^2_{iYt}	ρ_i
$(X_i, Y_i)_0$	535.0	104.70	1.0	0.6	0	1.00	0.60	0.00
$(X_i, Y_i)_{0.5}$	537.7	104.57	2.0	1.0	0	1.99	1.01	−0.071
$(X_i, Y_i)_1$	540.4	104.18	2.2	1.1	0	2.16	1.14	−0.130
$(X_i, Y_i)_{1.5}$	543.1	103.53	2.4	1.2	0	2.31	1.29	−0.185
$(X_i, Y_i)_2$	545.8	102.62	2.6	1.3	0	2.43	1.47	−0.231
$(X_i, Y_i)_{2.5}$	548.5	101.45	2.8	1.4	0	2.54	1.66	−0.266
$(X_i, Y_i)_3$	551.2	100.02	3.0	1.5	0	2.62	1.88	−0.293

TABLE 6.2: *Bivariate Gaussian probability distribution parameters (with respect to both the local and global frames of reference) corresponding to the kinematic data in Table 6.1.*

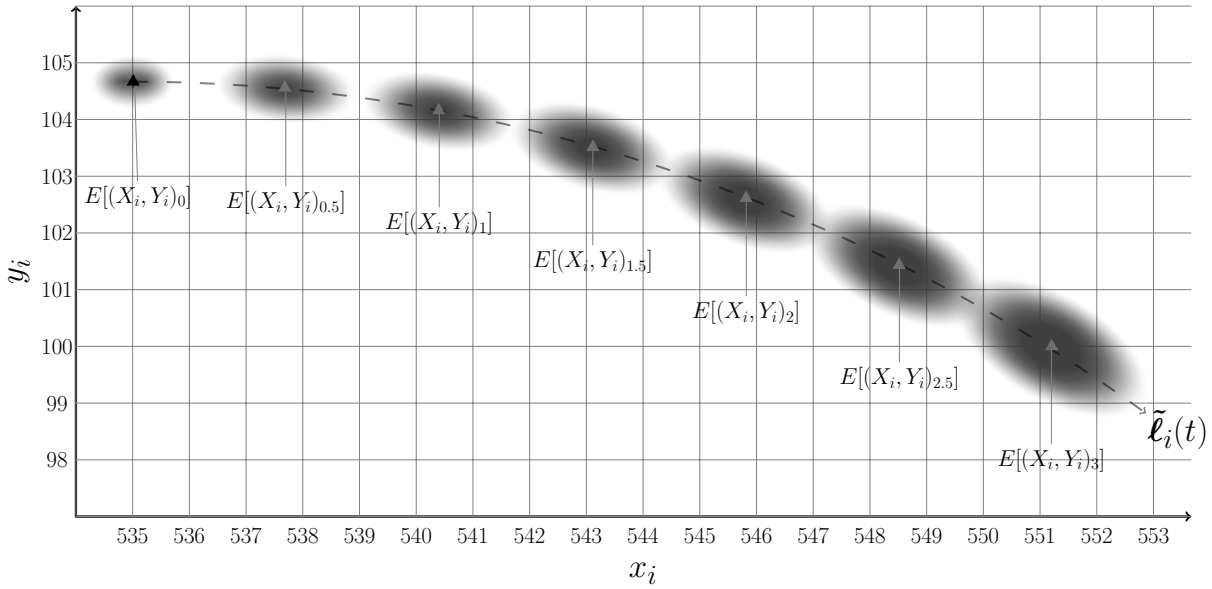


FIGURE 6.5: *Visualisation of the anticipated trajectory of VOI i over the time series of anticipated locations $(X_i, Y_i)_0, \ldots, (X_i, Y_i)_3$ in Table 6.2 with respective probability distributions $\mathcal{N}(\boldsymbol{\mu}_{i0}, \boldsymbol{\Sigma}_{i0}), \ldots, \mathcal{N}(\boldsymbol{\mu}_{i3}, \boldsymbol{\Sigma}_{i3})$.*

actually have to be calculated in the space-time continuum in the form of a shortest set of routing arcs. In other words, for the sake of practicality of MLE response selection computations, the parameter $0 \leq t \leq T$ should rather be defined on $\mathbb{R}$ in the time series $\{(X_i, Y_i)_t\}$, because MLE response selection computations are not possible based solely on the location distribution of an estimated scheduled interception time.

In §6.3.2, it was hinted that an anticipated VOI velocity vector may be obtained from a continuous mapping of anticipated VOI locations obtained from the anticipated trajectory (or anticipated system tracks) of the VOI calculated by MLE threat detection systems. This function of anticipated VOI locations is depicted in Figure 6.5 in the context of the worked example and is denoted by a vector $\tilde{\boldsymbol{\ell}}_i(t)$. It is thus assumed that the expected central tendencies $E[(X_i, Y_i)_t] = \tilde{\boldsymbol{\ell}}_i(t)$ are known for all $t \in [0, T]$.

In order to address the issue of expressing the anticipated position of a VOI as well as the level of uncertainty surrounding that anticipated position over a time continuum, however, define the *dynamic bivariate Gaussian probability distribution* of the random variables $(X_i, Y_i)_t$ as $(X_i, Y_i)_t \sim (\mu_{iX}(t), \mu_{iY}(t), \sigma^2_{iX}(t), \sigma^2_{iY}(t), \rho_i(t))$ or $(X_i, Y_i)_t \sim \mathcal{N}(\boldsymbol{\mu}_i(t), \boldsymbol{\Sigma}_i(t))$. Note the difference in notation from the discrete formulation $\mathcal{N}(\boldsymbol{\mu}_{it}, \boldsymbol{\Sigma}_{it})$, in the sense that this distribution

is now defined as a continuous function of time and. Here, as discussed above, the anticipated VOI location vector

$$\tilde{\boldsymbol{\ell}}_i(t) = (x_i(t), y_i(t)) = E[(X_i, Y_i)_t] = \boldsymbol{\mu}_i(t) = \begin{bmatrix} \mu_{iX}(t) \\ \mu_{iY}(t) \end{bmatrix}$$

may be thought of as an adequate estimator of the function of the central tendencies of this distribution. Moreover, the estimator functions $\breve{\sigma}^2_{iX}(t)$ and $\breve{\sigma}^2_{iY}(t)$ are defined similarly as in §6.3.4 to model the expected growth in uncertainty relative to the anticipated velocity vector of the VOI expressed in terms of temporary frames of reference, from where the distribution uncertainty parameters may be derived in canonical form. It follows from (6.2)–(6.4) that

$$\sigma^2_{iX}(t) = \breve{\sigma}^2_{iX}(t)\cos^2\theta_i(t) + \breve{\sigma}^2_{iY}(t)\sin^2\theta_i(t), \tag{6.5}$$

$$\sigma^2_{iY}(t) = \breve{\sigma}^2_{iX}(t)\sin^2\theta_i(t) + \breve{\sigma}^2_{iY}(t)\cos^2\theta_i(t) \tag{6.6}$$

and

$$\rho_i(t) = \frac{\sin\theta_i(t)\cos\theta_i(t)(\breve{\sigma}^2_{iX}(t) - \breve{\sigma}^2_{iY}(t))}{\sqrt{\left(\breve{\sigma}^2_{iX}(t)\cos^2\theta_i(t) + \breve{\sigma}^2_{iY}(t)\sin^2\theta_i(t)\right)\left(\breve{\sigma}^2_{iX}(t)\sin^2\theta_i(t) + \breve{\sigma}^2_{iY}(t)\cos^2\theta_i(t)\right)}}. \tag{6.7}$$

To demonstrate the working of this parametric multivariate distribution, reconsider the example of §6.3.5. Of course, by assumption,

$$\boldsymbol{\mu}_i(t) = \begin{bmatrix} 535 + 5.4t \\ 104.7 - 0.52t^2 \end{bmatrix}.$$

The variance functions in terms of the global frame of reference can, in essence, be estimated in two different ways. On the one hand, these noise functions may be expressed in the form of the continuous step functions

$$\breve{\sigma}^2_{iX}(t) = \begin{cases} 1 + 2t\,, & \text{if } 0 \le t \le 0.5 \\ 1.8 + 0.4t\,, & \text{if } 0.5 \le t \le 3 \end{cases}$$

and

$$\breve{\sigma}^2_{iX}(t) = \begin{cases} 0.6 + 0.8t\,, & \text{if } 0 \le t \le 0.5 \\ 0.9 + 0.2t\,, & \text{if } 0.5 \le t \le 3 \end{cases}$$

in terms of the local frame of reference. The dynamic covariance function with respect to the global frame of reference may then be computed according to (6.5)–(6.7). The second method consists of using polynomial interpolation in order to find a continuous function that best fits the estimated (global) variances derived in the discrete case. Using the values derived in Table 6.2, the variance function over the time interval $[0, 3]$ along the global x-axis may best be estimated by the fifth-degree polynomial

$$\sigma^2_{iX}(t) \approx 0.1t^5 - 0.92t^4 + 3.06t^3 - 4.77t^2 + 3.69t + 1,$$

while the variance function over the time interval $[0, 3]$ along the global y-axis may best be estimated by the fourth-degree polynomial

$$\sigma^2_{iY}(t) \approx -0.05t^4 + 0.39t^3 - 0.93t^2 + 1.15t + 0.6.$$

6.3.7 Confidence ellipses

As described earlier, a multivariate Gaussian probability distribution elliptic contour comprises a region of random variable value combinations (defined within an infinitesimally small area) so that each random variable has an identical chance of being observed. An elliptic contour is, however, mostly thought of as bounding a measurable set (in this case a probabilistic subspace measured in the interval $(0, 1)$) containing a specific subset of random variable value combinations distributed around the central tendencies of the underlying Gaussian distribution. These random variable values are then said to possess identical *Mahalanobis distances* [98] and, in cases where the random variables are uncorrelated, this metric may be simplified to a normalised Euclidean distance. These contours are often called *confidence ellipsoids* for practical reasons, or *confidence ellipses* in the bivariate case.

Formally speaking, α-bounded confidence ellipsoids represent values for the population parameter vector $\boldsymbol{\mu}_i$ for which the difference between these parameter values and the (*a posteriori*) observed estimate is not *statistically significant* at the $1 - \alpha$ % level [75], where α, typically defined in the interval $(0.80, 1)$, is the desired level of confidence prescribed by the decision maker. This probability bound may therefore be thought of as an indicator of the reliability of the estimation procedure rather than as some specific pre-calculated interval.

Without loss of generality, all time-related subscripts in this description of confidence ellipses are omitted from the parameters. For the sake of lucidity, it is beneficial to first analyse confidence ellipses mapped in the domain space of uncorrelated bivariate Gaussian random variables. In this simplified case, the lengths of the principal axes of the confidence ellipse may be defined solely in terms of the standard deviations σ_{iX}^2 and σ_{iY}^2. The equation of the ellipse is given by

$$\left(\frac{x_i - \mu_{iX}}{\sigma_{iX}^2}\right)^2 + \left(\frac{y_i - \mu_{iY}}{\sigma_{iY}^2}\right)^2 = \Phi_i,$$

where the parameter Φ_i represents a positive real number defining the *scale* of the ellipse with respect to the underlying Gaussian distribution. A question arises as to how this parameter should be configured so that the resulting elliptic contour precisely matches a pre-established confidence level α. Because the random variables X_i and Y_i are uncorrelated, the sum of the left hand side of the equation above represents the sum of squares of two independently Gaussian distributed random variables. It is known that the sum of squared Gaussian random variables is distributed according to the *Chi-Square probability distribution* [133], so it follows that the parameter Φ_i may be assessed accordingly by calculating the corresponding Chi-Square likelihood with *two* degrees of freedom (*i.e.* one for each unknown). Moreover, as suggested earlier, confidence ellipses may be described as the accumulated probabilities of all infinitesimally small regions contained inside this ellipse, and this is analogous to accumulating all points on these infinitely many ellipses of monotonically decreasing scales that are bounded above by the confidence ellipse of scale Φ_i. In other words, $P\left(\Phi_i < \chi_\alpha^2(2)\right) = \alpha$, where $\chi_\alpha^2(2)$ represents the upper critical value[4] of the Chi-Square distribution with two degrees of freedom and a confidence ellipse bounded by the parameter α. Informally speaking, an α% confidence interval may therefore be thought of as related to the value $\chi_\alpha^2(2)$ in such a way that α% of the distribution density is bounded inside the ellipse

$$\left(\frac{x_i - \mu_{iX}}{\sigma_{iX}^2}\right)^2 + \left(\frac{y_i - \mu_{iY}}{\sigma_{iY}^2}\right)^2 = \chi_\alpha^2(2).$$

Finally, by definition of the ellipse in the context of analytical geometry, the lengths of the principal axes of a confidence ellipse are $2\sigma_{iX}\sqrt{\chi_\alpha^2(2)}$ and $2\sigma_{iY}\sqrt{\chi_\alpha^2(2)}$.

[4] These values may be found in any Chi-Square probability tables [107].

The case for which the random variables are correlated is now considered. Here, the resulting confidence ellipses will not be axes-aligned with respect to the global reference coordinate frame. The eigenvectors of the covariance matrix[5] of a bivariate Gaussian distribution correspond to the principal directions of the distribution, while its eigenvalues correspond to the degrees of spread induced by the variances along the corresponding directions [107]. In other words, the eigenvectors of the covariance matrix of the underlying Gaussian distribution represent the direction of the error vectors along the largest spread of density points (namely along the major and minor axes of the ellipse), while the eigenvalues represent the magnitude of error with respect to the corresponding vectors. Numerically speaking, the lengths of the principal axes of the rotated confidence ellipse are given by $2\sqrt{\chi^2_\alpha(2)\lambda_1}$ and $2\sqrt{\chi^2_\alpha(2)\lambda_2}$, where λ_1 and λ_2 are the eigenvalues of the covariance matrix. For consolidation and validation purposes (with regard to the local frames of reference derived in §6.3.4), it is useful to observe at this point that the orientation of a correlated confidence ellipse with respect to the global x-axis is $\arctan(\boldsymbol{w}_{\lambda_1}\boldsymbol{w}_{\lambda_2}^{-1})$, where $\boldsymbol{w}_{\lambda_1}$ is the eigenvector of the covariance matrix corresponding to the largest eigenvalue (*i.e.* the eigenvalue associated with the major axis of the confidence ellipse) [133].

6.3.8 Discussion

The aim in this section was to propose a method for configuring stochastic input data pertaining to anticipated VOI locations into a framework that is both feasibly tractable from the standpoint of MLE threat detection systems and directly applicable to routing operations from the standpoint of MLE response selection systems. This was achieved in such a way that the output information of the former system forms a direct bridge with the input data of the latter. First, it was argued in §6.2 that it would be unrealistic to model the calculated interception points with VOIs in space deterministically. The bivariate Gaussian probability distribution was therefore introduced as a means of capturing the stochastic aspect of this calculation, and its adequacy to fulfill this task in the context of MLE response selection operations was motivated.

In §6.3.2 it was suggested that the central tendencies of the anticipated locations of a VOI at future points in time may be modelled within the current time stage by considering the points mapped by the anticipated trajectory derived from a pre-determined anticipated VOI velocity vector. A discussion on various factors that may contribute to the magnitude of the VOI locations Gaussian noise in an MLE environment was conducted in §6.3.3. Thereafter, it was demonstrated in §6.3.4 and §6.3.5 that it may be possible to express the uncertainty surrounding the expected location of a VOI at some point in the future by considering two perpendicular error vectors aligned with the major and minor axes of this anticipated VOI location in such a way as to consider the eigenvector corresponding to the major axis analogously to the (anticipated) heading vector of the VOI at that point in time *via* a local frame of reference. Because MLE response selection systems plan visitation routes along a time-continuum, discrete anticipated VOI location probability distributions may not be appropriate, leading to the definition of the dynamic bivariate Gaussian distribution in §6.3.6 in order to address this issue. The notion of confidence ellipses was finally introduced in §6.3.7 as a means to bound these bivariate Gaussian distributions from above. The concepts of the dynamic bivariate Gaussian distribution and confidence ellipses will be brought together later in the dissertation for the purpose of defining so-called *hyper-dimensional Gaussian corridors*. These corridors may be used to configure problem instance reinitialisation protocols (disturbance triggering conditions) pertaining to "significant" deviations in *actual* VOI velocity vectors relative to VOI velocity vectors anticipated *a priori*.

[5]The eigenvalues of a covariance matrix may easily be calculated from the familiar characteristic equation $det(\boldsymbol{\Sigma} - \boldsymbol{\lambda I}) = 0$.

As stressed earlier, the ability to estimate VOI trajectories with high accuracy is highly improbable. This may, however, not always be a significant shortcoming. If the progression of a VOI scheduled for visitation is close enough to its pre-calculated anticipated trajectory, for example, MLE resources can of course modify their bearings accordingly while travelling along their pre-assigned routing arcs. Only if a VOI changes velocity so unexpectedly or to such an extent that the currently implemented solution may significantly deteriorate from a macro-operational point of view should a disturbance be triggered.

6.4 Stochastic information pertaining to route distances

As discussed in §2.6.4 and in Chapter 4, a typical VRP objective evolves around the minimisation of planned route costs while ensuring that the chances of route durations/distances being in excess of the assigned vehicle's time/distance autonomy does not exceed a pre-determined threshold. In the presence of deterministic information pertaining to routing arc lengths and customer locations, these goals may be achieved relatively easily during the decision making process. In cases where such information is available probabilistically, however, the task may not be so easy.

Provided with a visitation route containing a subset of VOIs to be serviced, the interception points with these VOIs, the anticipated probability distributions of each of these VOIs at their expected times of interception (see §6.3.6), a specific MLE resource assigned to the route, and the autonomy threshold and current autonomy level of the MLE resource in question, an MLE response selection operator will ultimately be interested to know two things. First, what is the probability density function of the overall distance of the route. More specifically, as travelling costs carry most of the weight in the operating cost function (see Objective III in §4.2.3), the decision maker will clearly be interested in assessing the potential magnitudes of cost fluctuations (*i.e.* the size of the variances) associated with the combined set of routes spanning a solution. Based on the answers to these questions, the operator may decide whether the particular solution is worth implementing, given that the operating costs might, in some cases, be significantly larger than his *a priori* expectations[6]. Perhaps more importantly, modelling the operating costs associated with solutions by means of probability density functions allows for the formulation and evaluation of inter-solution dominance in stochastic multiobjective space (as will be seen later in the chapter). Secondly, the decision maker will be interested to gauge the probability that the autonomy levels of the allocated MLE resource expires whilst in the process of implementing a solution containing this route, in turn triggering an undesirable and perhaps avoidable route failure.

Mathematically speaking, consider the set of $n+2$ independent points $\{0, 1, \ldots, n, n+1\}$ such that the position of the first and last points, located in some global frame of reference, $\boldsymbol{z}_0 = (X_0, Y_0)$ and $\boldsymbol{z}_{n+1} = (X_{n+1}, Y_{n+1})$, are known with certainty, and such that the positions of the remaining n points, also located in the same global frame of reference, are all random variables modelled by the bivariate Gaussian distributions $\boldsymbol{z}_1 = (X_1, Y_1) \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $\boldsymbol{z}_2 = (X_2, Y_2) \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, $\ldots$, $\boldsymbol{z}_n = (X_n, Y_n) \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$. Without any loss of generality, let $(X_0, Y_0) \sim \mathcal{N}(\boldsymbol{z}_0^T, \boldsymbol{0})$ and $(X_{n+1}, Y_{n+1}) \sim \mathcal{N}(\boldsymbol{z}_{n+1}^T, \boldsymbol{0})$. Then, it is imperative to derive the joint probability density function of

$$\Theta(\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_n, \boldsymbol{z}_{n+1}) = \sum_{i=0}^{n} ||\boldsymbol{z}_i - \boldsymbol{z}_{i+1}||_2.$$

[6]Of course, it may also be the case that operating costs turn out to be less than expected.

6.4.1 On the probability distribution of route distances

Consider the *characteristic function*[7]

$$\phi_{\boldsymbol{z}}(\boldsymbol{\omega}) = E[e^{-\boldsymbol{\omega}\iota\boldsymbol{z}}] = e^{(\iota\,\boldsymbol{\omega}^T\boldsymbol{\mu} - \boldsymbol{\omega}^T\boldsymbol{\Sigma}\boldsymbol{\omega})}$$

of two random variables $(X, Y) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where ι is the imaginary unit $\sqrt{-1}$ and $\boldsymbol{\omega} \in \mathbb{R}^2$ is the so-called *argument vector* of the characteristic function. Moreover, consider the two independent[8] bivariate Gaussian random variables vectors $\boldsymbol{z}_i = (X_i, Y_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and $\boldsymbol{z}_j = (X_j, Y_j) \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, where $i, j \in \{0, 1, \ldots, n, n+1\}$ and $|i-j| = 1$. Then, considering that the sum of two independent random variables is, in fact, the product of their respective characteristic functions and acknowledging that the random variable $\boldsymbol{z}$ has the same covariance as the random variable $-\boldsymbol{z}$, but with opposite means [95], it follows that

$$\begin{aligned}
\phi_{(\boldsymbol{z}_i - \boldsymbol{z}_j)}(\boldsymbol{\omega}) &= e^{(\iota\,\boldsymbol{\omega}^T\boldsymbol{\mu}_i - \boldsymbol{\omega}^T\boldsymbol{\Sigma}_i\boldsymbol{\omega})} e^{(-\iota\,\boldsymbol{\omega}^T\boldsymbol{\mu}_j - \boldsymbol{\omega}^T\boldsymbol{\Sigma}_j\boldsymbol{\omega})} \\
&= e^{(\iota\,\boldsymbol{\omega}^T\boldsymbol{\mu}_i - \boldsymbol{\omega}^T\boldsymbol{\Sigma}_i\boldsymbol{\omega} - \iota\,\boldsymbol{\omega}^T\boldsymbol{\mu}_j - \boldsymbol{\omega}^T\boldsymbol{\Sigma}_j\boldsymbol{\omega})} \\
&= e^{(\iota\,\boldsymbol{\omega}^T(\boldsymbol{\mu}_i - \boldsymbol{\mu}_i) - \boldsymbol{\omega}^T(\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j)\boldsymbol{\omega})},
\end{aligned}$$

and so $(\boldsymbol{z}_i - \boldsymbol{z}_j) \sim \mathcal{N}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j)$. If $\boldsymbol{d} = (\boldsymbol{z}_i - \boldsymbol{z}_j) \sim \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$, then

$$\boldsymbol{\mu}_d = \begin{bmatrix} \mu_{iX} - \mu_{jX} \\ \mu_{iY} - \mu_{jY} \end{bmatrix}$$

and

$$\begin{aligned}
\boldsymbol{\Sigma}_d &= \begin{bmatrix} \sigma_{iX}^2 & \rho_i\sigma_{iX}\sigma_{iY} \\ \rho_i\sigma_{iX}\sigma_{iY} & \sigma_{iY}^2 \end{bmatrix} + \begin{bmatrix} \sigma_{jX}^2 & \rho_j\sigma_{jX}\sigma_{jY} \\ \rho_j\sigma_{jX}\sigma_{jY} & \sigma_{jY}^2 \end{bmatrix} \\
&= \begin{bmatrix} \sigma_{iX}^2 + \sigma_{jX}^2 & \rho_i\sigma_{iX}\sigma_{iY} + \rho_j\sigma_{jX}\sigma_{jY} \\ \rho_i\sigma_{iX}\sigma_{iY} + \rho_j\sigma_{jX}\sigma_{jY} & \sigma_{iY}^2 + \sigma_{jY}^2 \end{bmatrix}.
\end{aligned}$$

Moreover, the squared Euclidean distance between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ is given by $\boldsymbol{d}^T\boldsymbol{d} = (\boldsymbol{z}_i - \boldsymbol{z}_j)^T(\boldsymbol{z}_i - \boldsymbol{z}_j)$. But this also corresponds to the *inner product* $\langle \boldsymbol{d}, \boldsymbol{d} \rangle$ in $\mathbb{R}^2$ and, while the product of two bivariate Gaussian distributions may not be normalised as a density of the underlying vector of random variables [112], computing the probability distribution of this inner product is perfectly plausible since both distributions are identical and thus of the same dimension. The interested reader is referred to [77, 97, 112, 144] for discussions on the probability distributions of the product of multivariate Gaussian probability distributions. Given these identical distributions, the probability density functions may be expressed as

$$\begin{aligned}
\boldsymbol{d}^T\boldsymbol{d} &\sim \langle \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d), \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d) \rangle \\
&\sim \mathcal{N}(\boldsymbol{\Sigma}_d(\boldsymbol{\Sigma}_d + \boldsymbol{\Sigma}_d)^{-1}\boldsymbol{\mu}_d + \boldsymbol{\Sigma}_d(\boldsymbol{\Sigma}_d + \boldsymbol{\Sigma}_d)^{-1}\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d(\boldsymbol{\Sigma}_d + \boldsymbol{\Sigma}_d)^{-1}\boldsymbol{\Sigma}_d) \\
&\sim \mathcal{N}(2\boldsymbol{\Sigma}_d(2\boldsymbol{\Sigma}_d)^{-1}\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d(2\boldsymbol{\Sigma}_d)^{-1}\boldsymbol{\Sigma}_d).
\end{aligned}$$

This result suggests that the square distance between two Gaussian distributed bivariate random variables is itself, in fact, Gaussian (univariate, of course). Moreover, since there exists a one-to-one correspondence between the characteristic function and cumulative distribution functions

[7] A characteristic function provides an alternative way of describing a random variable. If a random variable is governed by a density function, then the characteristic function is its *dual* in the sense that each of these functions is a *Fourier transformation* of the other [75].

[8] With respect to one another, that is, but not necessarily with respect to the correlation status of their respective individual random variables.

of a random variable, it is always possible to derive one of these functions from the other one. According to Kettani and Ostrouchov [77], the characteristic function of this probability density function is given by

$$\phi_{\boldsymbol{d}^T\boldsymbol{d}}(\omega) = \frac{\exp\left(tr\{-\omega\iota(\boldsymbol{I}+2\omega\iota\boldsymbol{\Sigma}_d)^{-1}\boldsymbol{\mu}_d^T\boldsymbol{\mu}_d\}\right)}{\sqrt{|\boldsymbol{I}+2\omega\iota\boldsymbol{\Sigma}_d|}},$$

where $\omega \in \mathbb{R}$ is the argument of the characteristic function and the function $tr\{.\}$ represents the *trace* of a matrix (*i.e.* the sum of its main diagonal entries). Simplified probability density functions for special cases of the bivariate normal distribution are also derived in [77]. The random variable $\boldsymbol{d}^T\boldsymbol{d}$ has, for example, been shown to assume a *non-central Gamma distribution* in cases where $\boldsymbol{\Sigma}_d = \sigma^2\boldsymbol{I}$, a *Stacy distribution* in cases where $\boldsymbol{\Sigma}_d$ is diagonal and $\boldsymbol{\mu} = \boldsymbol{0}$, and a *Chi-square distribution* in cases where $\boldsymbol{\Sigma}_d = \boldsymbol{I}$ and $\boldsymbol{\mu} = \boldsymbol{0}$. It is, unfortunately acknowledged that the cumulative distribution functions of both random variables $\boldsymbol{d}^T\boldsymbol{d}$ and $\sqrt{\boldsymbol{d}^T\boldsymbol{d}}$ are most probably not available in closed form, calling for a rapid numerical evaluation of these complex expressions.

It would be unrealistic to assume that inter-arc distance random variables are independent (*i.e.* to assume that the sum of the distances linking all points $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{n+1}$ is distributed according to a linear combination of Gaussian distributions). The distance between two consecutive points $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ will, in fact, most certainly influence the distance between the two consecutive points $\boldsymbol{z}_j$ and $\boldsymbol{z}_k$. Let D_{ij} represent the squared Euclidean distance linking point $\boldsymbol{z}_i$ to point $\boldsymbol{z}_j$. Moreover, let the random variable D^k represent the total distance of the route travelled by MLE resource k. The density function of the random variable D^k may then be expressed as the *joint probability density function* $f_{D^k}(D^k) = f(d_{01}, d_{12}, \ldots, d_{n_k(n_k+1)})$ where, applying the rule of *prediction decomposition* [53] directly to a linear combination of multiple continuous conditional distributions,

$$\begin{aligned}
f_{D^k}(d^k) &= f_{D_{01}}(d_{01}) + \sum_{c=1}^{n_k} f(d_{c(c+1)}|d_{(c-1)c}, \ldots, d_{01}) \\
&= f_{D_{01}}(d_{01}) + \frac{f(d_{01}, d_{12})}{f_{D_{01}}(d_{01})} + \frac{f(d_{01}, d_{12}, d_{23})}{f(d_{12}|d_{01})f_{D_{01}}(d_{01})} + \frac{f(d_{01}, d_{12}, d_{23}, d_{34})}{f(d_{23}|d_{01}, d_{12})f(d_{12}|d_{01})f_{D_{01}}(d_{01})} \\
&+ \ldots + \frac{f(d_{01}, d_{12}, \ldots, d_{n_k(n_k+1)})}{\left(\prod_{s=1}^{n_k} f(d_{s(s+1)}|f(d_{01}, d_{12}, \ldots, d_{(s-1)s})\right) f_{D_{01}}(d_{01})} \\
&= f_{D_{01}}(d_{01}) + \frac{f(d_{01}, d_{12})}{f_{D_{01}}(d_{01})} + \sum_{c=2}^{n_k}\left(\frac{f(d_{01}, \ldots, d_{c(c+1)})(f_{D_{01}}(d_{01}))^{-1}}{\left(\prod_{s=2}^{c} f(d_{(s-1)s}|f(d_{01}, \ldots, d_{(s-2)(s-1)})\right)}\right).
\end{aligned}$$

Note that the density function of the random variable D_{01} is known in advance since its nature is not influenced by the natures of the distances of any other routing arcs. The cumulative distribution may therefore be expressed in terms of the accumulated conditional density functions as

$$\begin{aligned}
F_{D^k}(d^k) &= P(D^k \leq d^k) \\
&= \int_0^{d_{01}} f_{D_{01}}(\Lambda_{01})\, \mathrm{d}\Lambda_{01} + \int_0^{d_{01}}\int_0^{d_{12}} \frac{f(\Lambda_{01}, \Lambda_{12})}{f_{D_{01}}(\Lambda_{01})}\, \mathrm{d}\Lambda_{12}\Lambda_{01} + \\
&\sum_{c=2}^{n_k}\left(\int_0^{d_{01}} \cdots \int_0^{d_{n_k(n_k+1)}} \frac{f(\Lambda_{01}, \ldots, \Lambda_{c(c+1)})(f_{D_{01}}(\Lambda_{01}))^{-1}}{\left(\prod_{s=2}^{c} f(\Lambda_{(s-1)s}|f(\Lambda_{01}, \ldots, \Lambda_{(s-2)(s-1)})\right)} \mathrm{d}\Lambda_{n_k(n_k+1)} \ldots \Lambda_{01}\right).
\end{aligned}$$

Assessing the numerous joint probability density functions above is, unfortunately, another tedious aspect of the MLE response selection problem considered in this dissertation. It is

finally noted that all visitation routes are fortunately typically independent of one another (by definition), and so the density function of the overall travelling costs in Objective III in §4.2.3 may be redefined in a stochastic context as $\sum_{k\in\mathcal{V}^r} f_{D^k}(d^k)$. Moreover, as it is the only stochastic element in this objective function, the travelling costs will solely determine the noise in respect of the expected value of this objective function.

6.4.2 Confidence bounds on route distances

While the derivation of route density functions described in the previous section may be challenging, the problem of ensuring that an MLE resource does not travel more than a specific distance with a certain probability may be solved alternatively by employing confidence ellipses (see §6.3.7). In this section, a simple, generic methodology is described for managing the risk of route failures by employing confidence ellipses. One way of estimating such a risk involves plotting elliptic contours around the anticipated central tendencies of the VOI interception points $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{n_k}$ (scheduled along the visitation route assigned to MLE resource k), in order to generate an upper bound estimate on the route distance travelled at a certain level of confidence.

Using the principles laid down in §6.2.1, let the random variable $(X_i, Y_i)_{t^{i*}} \sim \mathcal{N}(\boldsymbol{\mu}_i(t^{i*}), \boldsymbol{\Sigma}_i(t^{i*}))$ model the anticipated location of VOI i at the estimated time of interception, t^{i*}. In addition, let $\varepsilon_{\alpha i}$ represent the α-bounded confidence ellipse mapped around the anticipated central tendencies of the point of interception with the VOI, and let $e_i(\alpha)$ be half the length of the major axis of this ellipse. Furthermore, redefine the distance $d_{i(i+1)k}$ as the expected distance of the shortest path that MLE resource may take to travel between the two consecutive ellipses $\boldsymbol{\varepsilon}_i(\alpha)$ and $\boldsymbol{\varepsilon}_{(i+1)}(\alpha)$ (where $i \in \{1, 2, \ldots, n_k - 1\}$). An illustrative example of such a configuration is shown in Figure 6.6. Finally, let ς_i represent the anticipated speed of the VOI around the estimated time of interception, and let Γ_k^* represent the maximum cruising speed of MLE resource k.

Suppose that the decision maker selects a confidence level of $\alpha\%$ when dealing with managing the risk of route failures in general. Then, an upper bound on the distance travelled by MLE resource k to complete this route is given by

$$P(D_k \leq d_k^*) = P\left(\sum_{j=0}^{n_k} d_{j(j+1)k} + \sum_{i=1}^{n_k} \Gamma_k^* \left(\frac{2\boldsymbol{\varepsilon}_{\alpha i}}{\Gamma_k^* - \varsigma_i}\right) \leq d_k^*\right) = \alpha.$$

Here, the random variable D_k represents the *a priori* estimated overall distance travelled and the indicator d_{k^*} may be chosen in such a way as to have all confidence ellipses bounded from above by $\alpha\%$ of the probability volume of the underlying distributions. In other words, d_{k^*} will provide the required upper bound in terms of the maximum distance that may be covered by MLE resource k when subject to ellipses of size α. It is acknowledged that such an upper bound represents an extreme worst case scenario, where the MLE resource enters each ellipse from one side of its major axis while each VOI is, meanwhile, located on the other side of the ellipse, cruising in the opposite direction and aligned with the major axis itself.

6.5 On the benefits of conservatively estimated service times

As mentioned previously, the expected duration and variation of the service time associated with any VOI may be very difficult to assess in real life. This is mainly because every VOI is unique — even those embodying the same type of threat. Additionally, every MLE resource and the way it operates in an active state may also significantly affect the way in which it services

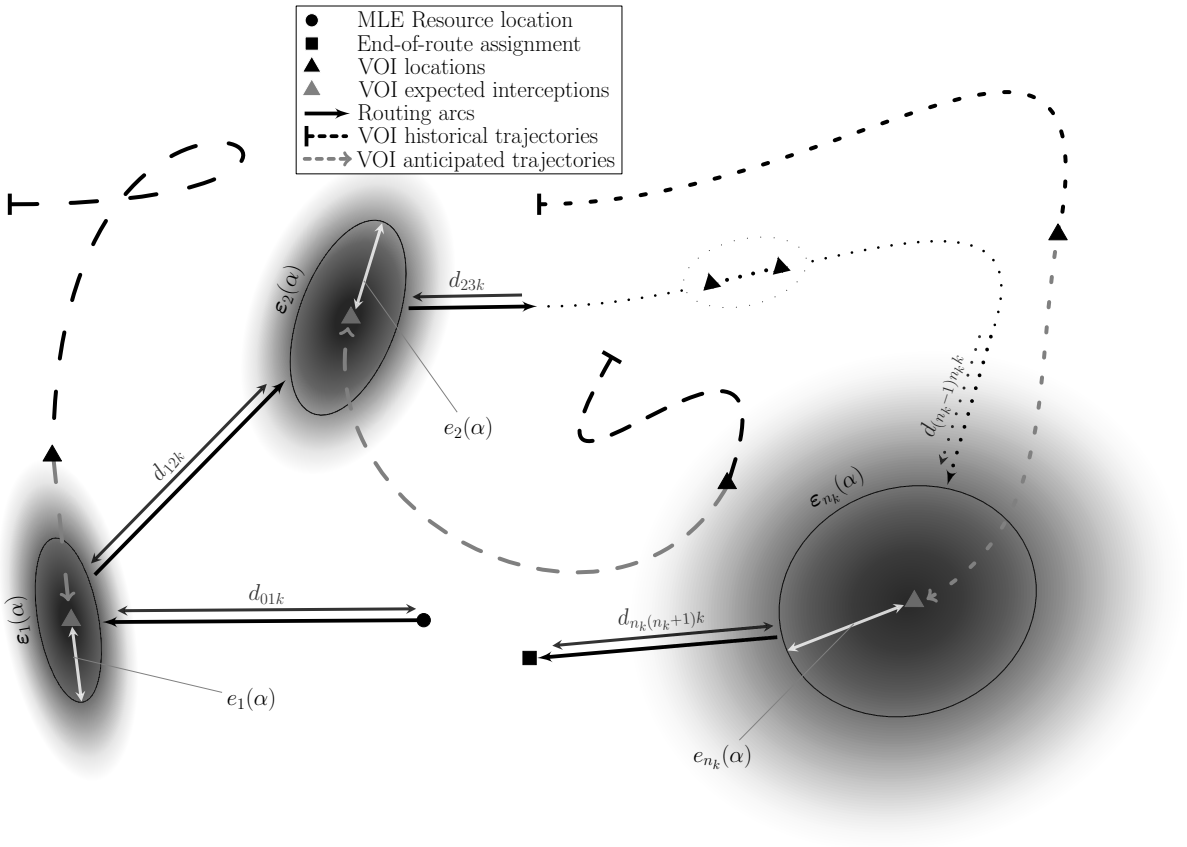


FIGURE 6.6: *Confidence ellipses plotted around the central tendencies of the VOI anticipated interception points along a visitation route may be employed as a means to manage the risk of route failures.*

VOIs. Even if appropriate probability distributions pertaining to VOI servicing times were to be derived, the noise accommodated around the central tendencies of these distributions will therefore most likely have to be impractically large in order to compensate for such uncertainty.

Fortunately, one of the unexpected scheduling benefits of the MLE response selection problem is related to the servicing time itself. Indeed, instead of painfully attempting to analyse the probability distributions of service times, the following seemingly obvious fundamental claim is made as a means of managing the uncertainty pertaining to VOI service times: *Assuming that human error is not present, the currently implemented solution will never deteriorate in quality or in feasibility whenever an MLE resource completes the service of a VOI under the pre-configured estimated service time of that encounter.*

This claim is primarily based on the following reasoning. In cases where an MLE resource takes longer than expected to service a VOI, it will have to intercept the remaining VOIs scheduled for visitation along its visitation route at a later point in time, hence potentially generating further delays. Incurring additional delays may, in most cases, also result in additional operating costs due to the MLE resource being required to travel further than originally anticipated distances in order to reach these remaining VOIs. In cases where the MLE resource completes service of a VOI earlier than expected, however, the MLE resource operator has the option to reconfigure its course and intercept the next VOI on its visitation route earlier than expected. In certain cases where the operator deems additional delay to be less important than incurring additional

operating costs, the MLE resource always has the option of engaging in a *wait-and-see* strategy after completing service earlier than expected. This approach may be considered in cases where the remaining travel arcs are expected to be cumulatively shorter as a result of waiting on the spot for some period of time. In closing, it is also acknowledged that the same reasoning may be applied to MLE resource setup times in cases where these are not deterministic.

6.6 Error management of threat evaluation input data

A proposed alternative to the implementation of Objective IV in §4.2.3 in the mathematical model formulation of the MLE response selection routing problem in respect of managing the risk of infeasible encounters is to employ pre-defined threshold parameter values. Such parameters will ensure that an MLE resource is forbidden to be scheduled to investigate a VOI associated with a probability above a certain threshold of embodying a certain type of threat. Hence, define the parameter $\theta_{kh} \in [0,1]$ to be an (upper bound) threshold associated with MLE resource k and threat type h. The constraint sets

$$p_{ih\tau} - \theta_{kh} \;\leq\; 1 - u_{kh\tau}$$

and

$$y_{ik\tau} \;\leq\; u_{kh\tau}, \qquad i \in \mathcal{V}^e_\tau, \qquad k \in \mathcal{V}^r,\ h \in \mathcal{H}$$

may then be incorporated into the model formulation as a means of implementing such an alternative towards lowering the overall risk of infeasible encounters, where $u_{kh\tau} \in \{0,1\}$ is a linking constraint variable. In the above constraint set formulation, MLE resource k is only authorised to intercept VOI i during time stage τ if $p_{ih\tau} \leq \theta_{kh}$ for all $h \in \mathcal{H}$. These constraints are only applicable to strictly incapable assignments; that is, there is no need to incorporate such constraints in respect of an MLE resource that it is not strictly incapable of handling a type of threat on its own.

The above approach may reduce the risk of infeasible encounters significantly when strict enough threshold parameters are established, and is easily implementable in the model formulation. Defining these threshold parameters in a subjective manner, however, may be a tedious process, and these parameters will most likely have to be reevaluated in the long run in order to accurately reflect the preferences of the coastal nation in that regard (for example, due to changes in the set of MLE resources, changes of operators, changes of threat intensity values and changes of national policies).

6.7 Dominance assessment in stochastic multiobjective spaces

The focus in this section shifts from considering stochastic elements in the domain space of a problem instance to its corresponding objective space. In the MLE response selection problem, two objective functions essentially ought to be defined probabilistically, namely the operating costs and the total delay times. On the one hand, operating costs mainly consist of the travelling costs of MLE resources which, as acknowledged in §6.4, may be highly stochastic. On the other hand, total delay times is even more highly stochastic, as it is itself determined by two stochastic elements, namely travel times and service times. It is assumed that threat evaluation systems already accommodate highly central probability distributions (*i.e.* small uncertainty in predictions) with respect to the potential threatening nature associated with each VOI in real time, as discussed in §4.1.4, and so visitation scores may be considered to be deterministic.

Due to the absence of literature on the matter, a generic approach is advocated toward defining criteria and conditions as a means of establishing a dominance relation between any two solutions mapped in the stochastic objective space of a multiobjective combinatorial optimisation problem. Of course, these generic principles may then be applied to the stochastic MLE response selection objectives mentioned above.

6.7.1 An introduction to the notion of "most likely better"

Consider a subset of m stochastic continuous objective functions $\tilde{\Theta} = \{f_1, \ldots, f_m\}$ in some K-objective combinatorial optimisation problem, and suppose that the complement subset of objective functions $\bar{\Theta} = \{f_{m+1}, \ldots, f_K\}$ are all deterministic. In addition, given any two decision vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, define the sets of random variables $\{O^1_{\boldsymbol{x}}, \ldots, O^m_{\boldsymbol{x}}\}$ and $\{O^1_{\boldsymbol{y}}, \ldots, O^m_{\boldsymbol{y}}\}$ as the stochastic objective function values of these solutions, respectively, where it is noted that two random variables $O^\ell_{\boldsymbol{x}}$ and $O^\ell_{\boldsymbol{y}}$ share the same domain space for any $\ell \in \{1, \ldots, m\}$. Then, it is apparent that the Pareto dominance relationship $\boldsymbol{x} \succ \boldsymbol{y}$ introduced in §2.1.4 is no longer valid on its own, which calls for an adequate alternative domination status criterion to be established. In other words, a more suited dominance measure ought to be implemented as a means of assessing whether a solution is deemed "better", "worse" or "indifferent" relative to another whilst evaluating solutions in a multiobjective optimisation problem in which one or more objective functions is defined non-deterministically. This relation, denoted by $\succsim$, is henceforth defined as the "most likely better" relation in such a way that

$$\boldsymbol{x} \succ \boldsymbol{y} \text{ } \textit{if and only if} \begin{cases} \boldsymbol{x} \succsim \boldsymbol{y} & \text{with respect to } \tilde{\Theta}, \\ \boldsymbol{x} \succ \boldsymbol{y} & \text{with respect to } \bar{\Theta}. \end{cases}$$

6.7.2 Formal assessment procedure

Without loss of generality, consider a multiobjective optimisation problem in which all objective functions are to be *maximised*, and define the relation $\gtrsim$ as "most likely larger." Ideally, the decision maker may primarily be interested in assessing the probability that the score of the stochastic objective function value of a certain solution is most likely larger than the same function value of a different solution. Moreover, a solution will, by definition, dominate another one with respect to the subset of stochastic objective functions $\tilde{\Theta}$ if all its objective function values are deemed most likely larger than the respective objective function values of the other solution. Hence,

$$O^\ell_{\boldsymbol{x}} \gtrsim O^\ell_{\boldsymbol{y}} \text{ } \textit{if and only if } \text{ } P(O^\ell_{\boldsymbol{x}} > O^\ell_{\boldsymbol{y}}) = P(O^\ell_{\boldsymbol{x}} - O^\ell_{\boldsymbol{y}} > 0) > 1 - \eta^*, \ell \in \{1, \ldots, m\},$$

where $\eta^* \in (0, 0.5)$ is a user-defined parameter reflecting the so-called *rigidity factor* associated with the claim that an objective function value is most likely larger than another. It is expected that the nature of the rigidity factor will have a significant impact on the development, nature, diversity and final size of the non-dominated front, and so it is imperative to perform various sensitivity analysis tests in order to select adequate configurations for this parameter.

Considering that two such random variables are independent to one another, probability calculations may be computed with relative ease. For example, in the context of the MLE response selection problem, it is expected that the Gamma and/or Gaussian probability distributions will be most suited to model the stochastic nature of the operating costs and delay times of visitation routes. The following two identities may therefore prove to be useful in this problem: (1) if $O_{\boldsymbol{x}} \sim \Gamma(a_{\boldsymbol{x}}, \theta)$ and $O_{\boldsymbol{x}} \sim \Gamma(a_{\boldsymbol{y}}, \theta)$, then $(O_{\boldsymbol{x}} - O_{\boldsymbol{y}}) \sim \Gamma(a_{\boldsymbol{x}} - a_{\boldsymbol{y}}, \theta)$, and (2) if $O_{\boldsymbol{x}} \sim \mathcal{N}(\mu_{\boldsymbol{x}}, \sigma^2_{\boldsymbol{x}})$

and $O_{\boldsymbol{x}} \sim \mathcal{N}(\mu_{\boldsymbol{y}}, \sigma^2_{\boldsymbol{y}})$, then $(O_{\boldsymbol{x}} - O_{\boldsymbol{y}}) \sim \mathcal{N}(\mu_{\boldsymbol{x}} - \mu_{\boldsymbol{y}}, \sigma^2_{\boldsymbol{x}} + \sigma^2_{\boldsymbol{y}})$. The computations may, however, become very tedious to carry out if the two random variables are governed by different probability distributions or if the random variables are defined non-parametrically, particularly considering that these calculations must be carried out rapidly during the course of a solution search process.

6.7.3 Alternative assessment procedures

In cases where the probability $P(O^{\ell}_{\boldsymbol{x}} > O^{\ell}_{\boldsymbol{y}})$ is not easily assessed, a subset of the following three proposed alternative stochastic dominance conditions may, alternatively, be employed:

1. $O^{\ell}_{\boldsymbol{x}} \gtrsim O^{\ell}_{\boldsymbol{y}}$ *if and only if* $P(O^{\ell}_{\boldsymbol{x}} < E[O^{\ell}_{\boldsymbol{y}}]) < \beta^*$,

2. $O^{\ell}_{\boldsymbol{x}} \gtrsim O^{\ell}_{\boldsymbol{y}}$ *if and only if* $P(O^{\ell}_{\boldsymbol{y}} > E[O^{\ell}_{\boldsymbol{x}}]) < \gamma^*$, and

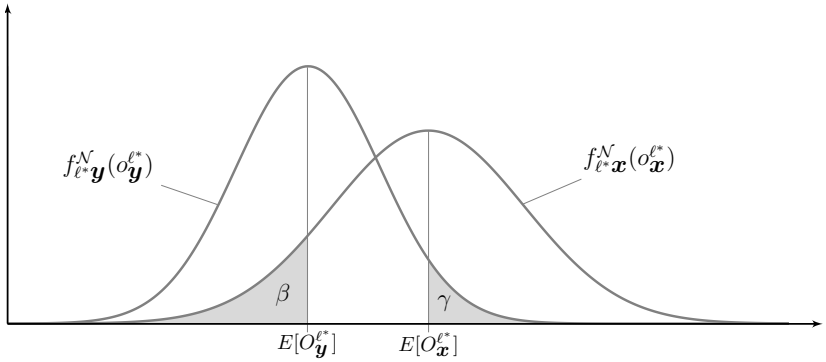
3. $O^{\ell}_{\boldsymbol{x}} \gtrsim O^{\ell}_{\boldsymbol{y}}$ *if and only if* $1 - \sqrt{\int_{-\infty}^{+\infty} \sqrt{f(o^{\ell}_{\boldsymbol{x}}) f(o^{\ell}_{\boldsymbol{y}})} d_{o^{\ell}}} < 1 - \zeta^*$.

In the first condition above, the probability that the objective function value $o^{\ell}_{\boldsymbol{x}}$ is greater than the central tendency (*i.e.* the expected value) of the objective function value $o^{\ell}_{\boldsymbol{y}}$, which must be larger than a user-defined parameter $\beta^* \in (0, 0.5)$ for the random variable $O^{\ell}_{\boldsymbol{x}}$ to be considered most likely larger than the random variable $O^{\ell}_{\boldsymbol{y}}$. The probability $P(O^{\ell}_{\boldsymbol{x}} < E[O^{\ell}_{\boldsymbol{y}}])$ is indicated graphically in Figure 6.7 by the shaded areas β.

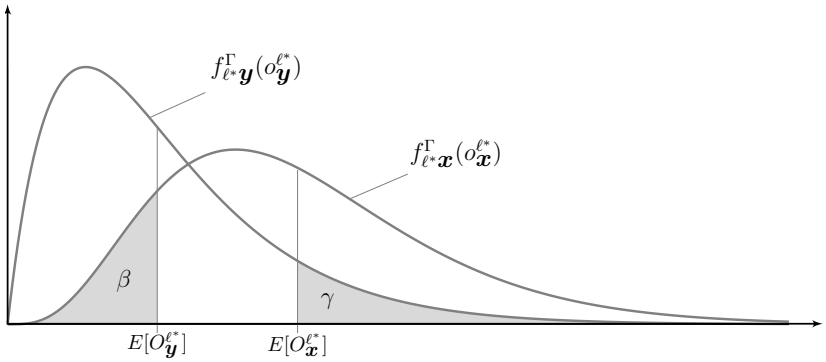
Similarly, the second condition above evaluates the probability that the objective function value $o^{\ell}_{\boldsymbol{y}}$ is greater than the central tendency of the objective function value $o^{\ell}_{\boldsymbol{x}}$, which must be smaller than a user-defined parameter $\gamma^* \in (0, 0.5)$ for the random variable $O^{\ell}_{\boldsymbol{x}}$ to be considered most likely larger than the random variable $O^{\ell}_{\boldsymbol{y}}$. The probability $P(O^{\ell}_{\boldsymbol{y}} > E[O^{\ell}_{\boldsymbol{x}}])$ is indicated graphically in Figure 6.7 by the shaded areas γ.

While these conditional approaches are relatively easily computed, they should, however, not be implemented on their own. Condition 1 does not take into consideration the magnitude of the variance of the random variable $O^{\ell}_{\boldsymbol{y}}$. As a result, although it may appear as if there exists a convincing case that $O^{\ell}_{\boldsymbol{x}}$ is larger than $E[O^{\ell}_{\boldsymbol{y}}]$, it may also very well be the case that a large (unacknowledged) probability γ is present, in turn indicating that there exists a significant counter-possibility that $O^{\ell}_{\boldsymbol{y}}$ is larger than $E[O^{\ell}_{\boldsymbol{x}}]$. A similar reasoning is applicable to Condition 2. It is thus recommended that Conditions 1 and 2 be employed simultaneously rather than on their own, noting, however, that such a union requires two user-defined parameters.

Lastly, although much more challenging to compute, one may also consider the so-called modified *Hellinger distance* [98] underpinned by Condition 3 above. This metric is defined on the interval $(0, 1)$ and agrees with the triangle inequality. Moreover, it is indicative enough to be employed on its own and only requires one user-defined parameter (namely ζ^*). A drawback of implementing this measure, however, arises from the possibly high level of complexity involved in assessing the product of the two underlying probability distributions. From a visual point of view, the (non-modified) Hellinger distance of two probability distributions sharing the same domain space is, in most cases, reflected by the surface area (in the univariate case) of the overlapping region between these two distributions [96].



(a) *Objective function mappings governed by Gaussian probability distributions.*



(b) *Objective function mappings governed by Gamma probability distributions.*

FIGURE 6.7: *Mappings of two decision vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ by stochastic objective function ℓ^* according to two types of probability distributions $f^{\mathcal{N}}$ and f^{Γ}.*

6.7.4 A numerical example

To illustrate the workings of the stochastic dominance measures proposed above, a basic numerical example is provided in this section[9]. Suppose that the operating costs $O^{costs}_{\boldsymbol{x}}$ (noting that this objective is to be *minimised* on the domain $\mathbb{R}^+$) associated with solution $\boldsymbol{x}$ may be modelled probabilistically by a Gaussian distribution with mean \$3 500 and standard deviation \$90, and suppose that the operating costs $O^{costs}_{\boldsymbol{y}}$ associated with solution $\boldsymbol{y}$ may be modelled probabilistically by a Gaussian distribution with mean \$3 700 and standard deviation \$125. That is, let $O^{costs}_{\boldsymbol{x}} \sim \mathcal{N}(3\,500, 8\,100)$ and $O^{costs}_{\boldsymbol{y}} \sim \mathcal{N}(3\,700, 15\,625)$.

The formal assessment procedure of §6.7.2 is first considered. It follows directly from the observations made in §6.4 for the random variable $Z = O^{costs}_{\boldsymbol{y}} - O^{costs}_{\boldsymbol{x}}$ that $Z \sim \mathcal{N}(200, 23\,725)$. Using the software package MATLAB [147],

$$P(O^{costs}_{\boldsymbol{x}} < O^{costs}_{\boldsymbol{y}}) = P(Z > 0) = \int_0^{\infty} f_{\mathcal{N}}(z)\mathrm{d}_z = 0.903.$$

By definition, $O^{costs}_{\boldsymbol{x}} \lesssim O^{costs}_{\boldsymbol{y}}$ provided that the rigidity factor satisfies $\eta^* > 0.097$. The domi-

[9]For the sake of clarity, some of the symbols in this example are simplified from their formal definitions.

nance measure pertaining to Condition 1 in §6.7.3 is given by

$$P(O_{\boldsymbol{x}}^{costs} > E[O_{\boldsymbol{y}}^{costs}]) = \int_{3\ 700}^{\infty} f_{\mathcal{N}}(o_{\boldsymbol{x}}^{costs})\mathrm{d}_{o^{costs}} \approx 0.013,$$

while the dominance measure pertaining to Condition 2 in §6.7.3 is given by

$$P(O_{\boldsymbol{y}}^{costs} < E[O_{\boldsymbol{x}}^{costs}]) = \int_{-\infty}^{3500} f_{\mathcal{N}}(o_{\boldsymbol{y}}^{costs})\mathrm{d}_{o^{costs}} \approx 0.045.$$

Finally, the modified Hellinger distance of §6.7.3 is assessed. According to Bromiley [15], the distribution of the product of two independent Gaussian random variables is given by

$$f_{\mathcal{N}\boldsymbol{x}}(o)f_{\mathcal{N}\boldsymbol{y}}(o) = \frac{\psi_{xy}}{\sqrt{2\pi\frac{\sigma_{\boldsymbol{x}}^2\sigma_{\boldsymbol{y}}^2}{\sigma_{\boldsymbol{x}}^2+\sigma_{\boldsymbol{y}}^2}}}\ \exp\left[-\frac{\left(w-\frac{\mu_{\boldsymbol{x}}\sigma_{\boldsymbol{y}}^2+\mu_{\boldsymbol{y}}\sigma_{\boldsymbol{x}}^2}{\sigma_{\boldsymbol{x}}^2+\sigma_{\boldsymbol{y}}^2}\right)^2}{2\frac{\sigma_{\boldsymbol{x}}^2\sigma_{\boldsymbol{y}}^2}{\sigma_{\boldsymbol{x}}^2+\sigma_{\boldsymbol{y}}^2}}\right],$$

which is clearly governed by a scaled Gaussian probability density function, where

$$\psi_{xy} = \frac{1}{\sqrt{2\pi(\sigma_{\boldsymbol{x}}^2+\sigma_{\boldsymbol{y}}^2)}}\ \exp\left[-\frac{(\mu_{\boldsymbol{x}}-\mu_{\boldsymbol{y}})^2}{2(\sigma_{\boldsymbol{x}}^2+\sigma_{\boldsymbol{y}}^2)}\right]$$

is itself a Gaussian probability density function modelling the random variable $\mu_{\boldsymbol{x}} - \mu_{\boldsymbol{y}}$ with mean 0 and standard deviation $\sqrt{\sigma_{\boldsymbol{x}}^2+\sigma_{\boldsymbol{y}}^2}$. Letting $f_{\mathcal{N}\boldsymbol{x}}(o)f_{\mathcal{N}\boldsymbol{y}}(o) = f_W(w)$, the density function may be written as

$$\begin{aligned}
f_W(w) &\approx \frac{e^{-\frac{4\ 000}{47\ 450}}}{\sqrt{2\pi}\sqrt{5\ 335}\sqrt{2\pi}\sqrt{23\ 725}}\ \exp\left[-\frac{(w-3\ 568)^2}{2(5\ 335)}\right] \\
&\approx \frac{0.919}{\sqrt{2\pi}\ 28\ 200}\ \exp\left[-\frac{(w-3\ 568)^2}{2(5\ 335)}\right] \\
&\approx \frac{1}{\sqrt{2\pi}\ 30\ 685}\ \exp\left[-\frac{(w-3\ 568)^2}{2(5\ 335)}\right] \\
&\approx 0.00238\left(\frac{1}{\sqrt{2\pi}\sqrt{5\ 335}}\ \exp\left[-\frac{(w-3\ 568)^2}{2(5\ 335)}\right]\right),
\end{aligned}$$

so that $W \sim 0.00238\,(\mathcal{N}(3\ 568, 5\ 335))$. Next, it is required to assess the density function of the random variable $V = \sqrt{W}$. Before going any further, noting that $V, W \in [0, \infty)$ and that the mapping of this function is both bijective and monotonically increasing, it is anticipated that $F_V(\infty) = 0.00238(F_W(\infty)) = 0.00238$. For the sakes of interest and validation, however, observe that $F_V(v) = P(V \leq v) = P(\sqrt{W} \leq v) = P(W \leq v^2) = F_W(v^2)$. Applying the chain rule for differentiation, the density function $f_V(v)$ may be formulated as

$$f_V(v) = \frac{\mathrm{d}}{\mathrm{d}v}F_V(v) = f_W(v^2)\frac{\mathrm{d}}{\mathrm{d}v}v^2 = 2vf_W(v^2).$$

Using the software package MATLAB [147], it follows that

$$\begin{aligned}
\int_0^{\infty} f_V(v)\,\mathrm{d}v &= \frac{0.00238}{\sqrt{2\pi}\sqrt{5\ 335}}\int_0^{\infty} 2v\ \exp\left[-\frac{(v^2-3\ 568)^2}{10\ 670}\right]\mathrm{d}v \\
&\approx \frac{0.00238}{\sqrt{2\pi}\sqrt{5\ 335}}(183.0868) \\
&\approx 0.00238,
\end{aligned}$$

which agrees with the above anticipation. The modified Hellinger distance between these two probability distributions is therefore estimated at $1 - \sqrt{0.00238} \approx 0.9512$.

6.8 Chapter summary

In the first section of this chapter, a thorough description of the stochastic elements of the MLE response selection problem was provided. In §6.2, an investigation was launched into effective ways of modelling the uncertainty pertaining to the position of VOIs in space, and it was argued that a bivariate Gaussian probability distribution is generally expected to work well in this respect. A step-by-step description of modelling the uncertainty associated with VOI visitation locations was then performed in §6.3, and this was followed by analysis of the probability density nature of visitation routes in §6.4. In §6.5 and §6.6, the stochastic information pertaining to VOI service times and VOI threatening natures were addressed, respectively. Finally, the focus shifted in §6.7 to the task of assessing various dominance measures in order to evaluate solutions to multiobjective optimisation problems containing one or more stochastic objective.

The next step is to propose a solution methodology for solving the MLE response selection problem. In particular, appropriate solution search techniques for solving instances of this complex problem are suggested in the next chapter.

CHAPTER 7

Optimisation Methodology

Contents

The MLE response selection problem is quite unique in its complexity in relation to other VRPs, as it has, *inter alia*, a heterogeneous fleet of vehicles, multiple depots, customer profits, multiple objectives (up to five objectives), uncertain or incomplete sources of input data and general system dynamism as its main characteristics. It is therefore critical, with respect to the overall MLE response selection system effectiveness, to design solution techniques that are capable of presenting the decision maker with a satisfactory set of non-dominated solutions within a limited budget of time. In this dynamic routing problem, the amount of time required to find a new, preferred solution increases the MLE resource response times (consequently having a negative indirect impact on Objective II of §4.2.3), while increasing the chances of detours (consequently having a negative indirect impact on Objective III), or reaching some VOIs too late (consequently having a negative indirect impact on Objective I). A solution search engine that can *quickly* generate *high-quality* approximations of the Pareto front is therefore preferred over one that *slowly* generates *Pareto-optimal* solutions, as it is believed that the former technique will, in all likelihood, still outperform decisions made by a human operator by a large margin in the

same amount of time. Due to its single-objective nature and relatively low complexity, solution methodologies for solving the VOI distribution process, as described in §4.4, are not presented in this chapter.

Many features associated with the VRP considered in this dissertation are ultimately absent in the literature on VRPs, particularly with respect to the implementation of customer visitation profits in a multiobjective optimisation context. Consequently, only certain aspects of the VRP foundations discussed in §2.6 may be leveraged in the design of optimisation methodologies for solving the MLE response selection problem, while the remaining parts are required to be derived anew.

This chapter is structured as follows. An algebraic approach towards calculating the interception point of two moving objects in a two-dimensional plane is derived in §7.1 and this is followed by a proposed solution approach in §7.2 for assigning MLE resources to patrol circuits and bases at the end of their missions. In §7.3, a functional and effective solution data encoding scheme (adapted especially for this problem) is proposed, while a formal description of a Pareto-front approximation is given in §7.4. Solution methodologies based on the method of simulated annealing and on the notion of a genetic algorithm (as well as a hybrid combining these two metaheuristics) are then proposed in §7.5–7.7. Two versions of the method of simulated annealing applied in the context of parallel computing are finally presented in §7.8, after which the chapter closes with a brief summary in §7.9.

7.1 Interception estimate of two objects moving on a 2D plane

Estimating interception trajectories is a pre-requisite for evaluating arc travel distances and arc travel times as part of objective function and constraint evaluations during the solution configuration process for the MLE response selection problem. Interception trajectory calculations are therefore used (1) to point an (active) MLE resource in the right direction based on its speed capability, the velocity of the VOI towards which it is heading, as well as various environmental vectors, and (2) to estimate the arc distances and travel times along visitation routes during the solution search process in order to evaluate a particular solution in terms of objective function performance and distance and time-constrained feasibility. It is critical to note that, in certain cases, there might not be any interception point available, particularly in cases where the VOI is close to the jurisdiction area boundary or to the coast (and may therefore reach either one before being intercepted) or when there exists no MLE resource fast enough or close enough to catch up with it. Such cases may, however, be dealt with in a subjective manner by the operator during the model management procedure.

This section contains an algebraic description of a method for approximating such interception points. It is assumed that an external environmental vector may be expressed as a velocity vector so as to take into consideration the effects of ocean currents and wind vectors. Furthermore, it is assumed that there are no obstructions in the way of the MLE resource while it is travelling towards the VOI. Calculating interception points by means of linear or non-linear parametric equations (as in the dynamically anticipated VOI velocity vector formulations of §6.3.2) falls outside the scope of this dissertation. Instead, a simplified approach is adopted where, given the known current geographical positions of an MLE resource and a VOI at sea, the anticipated movement of the VOI is assumed to be governed by a static velocity vector (see §6.3.2), while the MLE resource is assumed to travel at a fixed average speed over any given distance. It is then required to calculate a point of intercept between these two objects in a Cartesian plane so that the distance travelled by the MLE resource (and analogously the time taken for both

objects to reach that point) is minimised. This problem can be solved in constant time (*i.e.* no iterative steps are necessary).

As described in the previous chapter, the oceanic open plane staging the MLE response selection operations of a coastal nation may be expressed as a bounded two-dimensional space with global basis $\{\boldsymbol{b}_x, \boldsymbol{b}_y\}$. Under the above-mentioned assumptions, let the position of the VOI at time $t = 0$ be denoted by $\boldsymbol{P}_\kappa = [x_\kappa \; y_\kappa]$ and let its velocity vector be denoted by $\boldsymbol{v}_\kappa = [s_{\kappa x} \; s_{\kappa y}]$, where $s_{\kappa x}$ is the velocity component of the VOI in the direction of the basis vector $\boldsymbol{b}_x$ and $s_{\kappa y}$ is the velocity component of the VOI in the direction of the basis vector $\boldsymbol{b}_y$. Furthermore, let η be the average achievable speed of the scheduled MLE resource (which is assumed to be fixed over any distance), let $\boldsymbol{P}_\omega = [x_\omega \; y_\omega]$ be the position of the MLE resource at time $t = 0$ and denote the (unknown) velocity vector of the MLE resource by $\boldsymbol{v}_\omega = [s_{\omega x} \; s_{\omega y}]$, where $s_{\omega x}$ is the velocity component of the MLE resource in the direction of the basis vector $\boldsymbol{b}_x$ and $s_{\omega y}$ is the velocity component of the MLE resource in the direction of the basis vector $\boldsymbol{b}_y$. Moreover, define the environmental velocity vector $\boldsymbol{v}_\zeta = [s_{\zeta x} \; s_{\zeta y}]$ to embody all outside forces impacting the motion of objects at sea, such as current vectors and/or wind vectors. Finally, let the (unknown) coordinates $\boldsymbol{P'} = [x' \; y']$ represent the expected point of interception in space of these two objects and let $t' \geq 0$ be the time at which these two objects are expected to reach each other. These vector components are presented graphically in Figure 7.1.

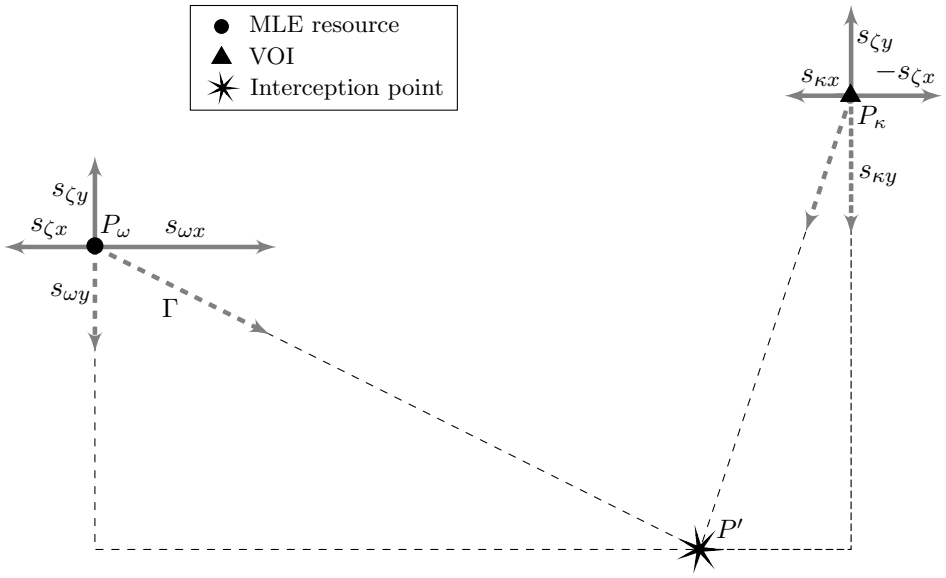


FIGURE 7.1: *Computing the point of interception for two objects at sea.*

In order to solve this interception problem, it is only necessary to solve for the unknowns $s_{\omega x}$, $s_{\omega y}$ and t'. The interception coordinates of $\boldsymbol{P'}$ are then given by

$$x' = x_\kappa + (s_{\kappa x} + s_{\zeta x})\, t' \quad \text{and} \quad y' = y_\kappa + (s_{\kappa y} + s_{\zeta y})\, t'. \tag{7.1}$$

Additionally, the interaction between the norm of the velocity vector of the VOI and the speed of the resource at the point of intersection, with respect to time, may be described as

$$||\boldsymbol{P}_\kappa + (\boldsymbol{v}_\kappa + \boldsymbol{v}_\zeta)\, t'||_2 = \|\boldsymbol{P}_\omega + (\boldsymbol{v}_\omega + \boldsymbol{v}_\zeta)\, t'\|_2, \tag{7.2}$$

where $||\cdot||_2$ denotes the well-known *L_2-norm*. Applying the definition of the L_2-norm in a Cartesian plane, (7.2) may be rewritten as

$$(x_\kappa + (s_{\kappa x} + s_{\zeta x})\, t')^2 + (y_\kappa + (s_{\kappa y} + s_{\zeta y})\, t')^2 = (x_\omega + (s_{\omega x} + s_{\zeta x})\, t')^2 + (y_\omega + (s_{\omega y} + s_{\zeta y}\, t')^2. \tag{7.3}$$

Expanding (7.3), and expressing the velocity components of the MLE resource in terms of its speed η yields the quadratic equation $At'^2 + Bt' + C = 0$, where

$$\begin{aligned} A &= {s_{\kappa x}}^2 + 2s_{\kappa x}s_{\zeta x} + {s_{\zeta x}}^2 - 2s_{\omega x}s_{\zeta x} - {s_{\zeta x}}^2 + {s_{\kappa y}}^2 + 2s_{\kappa y}s_{\zeta y} + {s_{\zeta y}}^2 - 2s_{\omega y}s_{\zeta y} - {s_{\zeta y}}^2 - \eta^2, \\ B &= 2x_\kappa(s_{\kappa x} + s_{\zeta x}) - 2x_\omega(s_{\omega x} + s_{\zeta x}) + 2y_\kappa(s_{\kappa y} + s_{\zeta y}) - 2y_\omega(s_{\omega y} + s_{\zeta y}) \text{ and} \\ C &= {x_\kappa}^2 - {x_\omega}^2 + {y_\kappa}^2 - {y_\omega}^2. \end{aligned}$$

Moreover, the pair of equations

$$x_\omega + (s_{\omega x} + s_{\zeta x})\,t' = x_\kappa + (s_{\kappa x} + s_{\zeta x})\,t'$$

and

$$y_\omega + (s_{\omega y} + s_{\zeta y})\,t' = y_\kappa + (s_{\kappa y} + s_{\zeta y})\,t'$$

hold, which may be rewritten as

$$s_{\omega x} = \frac{x_\kappa - x_\omega}{t'} + s_{\kappa x} \tag{7.4}$$

and

$$s_{\omega y} = \frac{y_\kappa - y_\omega}{t'} + s_{\kappa y}, \tag{7.5}$$

respectively. The quadratic equation $At'^2 + Bt' + C = 0$, together with (7.4) and (7.5), yield a system of three simultaneous equations allowing for the computation of the required three unknowns. Substituting $s_{\omega x}$ and $s_{\omega y}$ from (7.4) and (7.5) into this quadratic equation yields the alternative quadratic equation $\tilde{A}t'^2 + \tilde{B}t' + \tilde{C} = 0$, where

$$\begin{aligned} \tilde{A} &= {s_{\kappa x}}^2 + {s_{\kappa y}}^2 - \eta^2, \\ \tilde{B} &= 2(x_\kappa - x_\omega)(s_{\kappa x} + s_{\zeta x}) - 2s_{\zeta x}(x_\kappa - x_\omega) + 2(y_\kappa - y_\omega)(s_{\kappa y} + s_{\zeta y}) - 2s_{\zeta y}(y_\kappa - y_\omega) \text{ and} \\ \tilde{C} &= {x_\kappa}^2 - {x_\omega}^2 - 2x_\omega(x_\kappa - x_\omega) + {y_\kappa}^2 - {y_\omega}^2 - 2y_\omega(y_\kappa - y_\omega). \end{aligned}$$

The roots of this equation may be computed using the well-known quadratic formula to find the time of interception t', after which the coordinates of the interception point may be found from (7.1), as well as the MLE resource velocity vector components $s_{\omega x}$ and $s_{\omega y}$ from (7.4) and (7.5). Given its constant speed η and its velocity vector $\boldsymbol{v}_\omega$, the bearing towards which the MLE resource must head in order to intercept the VOI in the shortest path possible is then easily calculated using basic trigonometry.

7.2 End-of-route assignments

In this section, a method is proposed for allocating active MLE resources to vertices at the end of their routes. Although relatively simple, the proposed method may easily be incorporated as a sub-process of the optimisation search engine proposed later in this chapter and is not computationally expensive. In addition, this method ensures that an MLE resource remains within certain boundaries while carrying out its mission and, more importantly, ensures that no MLE resource is assigned a distance and/or time constrained infeasible route.

In this process, it is assumed that an MLE resource will always be scheduled to join a patrol circuit at the end of its route, provided that there exists at least one such candidate circuit, and provided that it is feasibly reachable. Otherwise it will relocate to one of the approved bases, provided that there exists at least one such candidate base and provided that it is feasibly reachable. In §4.2.1, so-called patrol autonomy thresholds were introduced in the mathematical modelling process, which ensure that an MLE resource is only allowed to join a patrol circuit after

completing its mission provided that the travel distance and travel time to that patrol circuit are within certain autonomy levels. In addition, it was proposed in §5.2.3 that the idle MLE resource management operator should provide some form of input to the MLE response selection operator with respect to end-of-route assignment preferences. Such input may be configured as a set of preferred destinations associated with each active MLE resource, represented as sets containing one or more elements from the patrol circuit and base sets.

The proposed end-of-route assignment methodology for an active MLE resource henceforth consists of the following steps: For each patrol circuit in $\mathcal{V}^p_{k\tau}$, the estimated distance and time autonomy level of MLE resource k are assessed if it were to travel to that patrol circuit after completing its mission. These autonomy levels are calculated by subtracting the expected distance and time that the MLE resource takes to travel to, and service, all VOIs assigned to its visitation route as well as covering the last arc along its route (*i.e.* the arc linking the last VOI on its visitation route to the patrol circuit) from respectively its distance and time autonomy levels $a^d_{k\tau}$ and $a^t_{k\tau}$ of that MLE resource at the beginning of time stage τ. If there exists at least one candidate patrol circuit $\rho \in \mathcal{V}^p_{k\tau}$ that is both within the distance and time patrol autonomy threshold parameters $A^d_{k\rho}$ and $A^t_{k\rho}$ of that MLE resource, then the patrol circuit at the closest distance from the MLE resource at the end of its mission is selected as its end-of-route vertex during time stage τ. If no such patrol circuit exists, then, for each base $b \in \mathcal{V}^b_{k\tau}$ (assuming that $|\mathcal{V}^b_{k\tau}| \geq 1$), the estimated distance and time autonomy level of the MLE resource is assessed if it were to travel to that base after completing its mission during time stage τ. If there exists at least one candidate base that is both within the distance and time autonomy levels of the MLE resource in question, then the base at the closest distance from the MLE resource at the end of its mission is selected as its end-of-route vertex. If no such base exists, then the route, and therefore, the entire solution, is classified as autonomy-constraint infeasible.

The reason for selecting end-of-route vertices based on spatial proximity from the last VOI in the visitation route of an MLE resource is motivated by the idea that the length of the last arc within the visitation route only impacts on travel costs (the travelling time associated with the last arc, on the other hand, has no impact on any of the objective function values). A pseudo-code description of the above end-of-route assignment procedure is given in Algorithm 7.1. Here, the term $EndofRoute_{k\tau}$ refers to the end-of-route vertex assigned to MLE resource k during time stage τ. Furthermore, the parameters

$$\tilde{a}^d_{k\tau} = a^d_{k\tau} - \eta_k \left(t_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)k\tau} - T_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\tau} \right)$$

and

$$\tilde{a}^t_{k\tau} = a^t_{k\tau} - t_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)k\tau} - T_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\tau}$$

are respectively the estimated distance and time autonomy levels of MLE resource k as soon as it has finished servicing the last VOI along its visitation route during time stage τ.

7.3 Solution representation and overall transformation process

Solutions to an MLE response selection problem instance should be encoded in very specific data formats which allow for the effective application of global and local search operations, easy evaluation of objective function values, end-of-route assignments and tests for solution feasibility. The various complex dynamic features associated with this problem, however, make it difficult to standardise these data formats to be used as part of an optimisation search process. Candidate solutions must be encoded in a suitably versatile format so as to be employed by different model components in a generic manner. In particular, the solution format should accommodate

Algorithm 7.1: End-of-route assignments procedure

Input : Initial autonomy levels $a^d_{k\tau}$ and $a^t_{k\tau}$, patrol autonomy thresholds $A^d_{k\rho}$ and $A^t_{k\rho}$, and model management input sets $\mathcal{V}^p_{k\tau}$ and $\mathcal{V}^b_{k\tau}$.

Output: End-of-route assignments for every active MLE resource during time stage τ.

forall $k \in \mathcal{V}^r$ **do**
  $EndofRoute_{k\tau} \leftarrow \emptyset$
  if $|\mathcal{V}e_{k\tau}| \geq 1$ **then**
    if $|\mathcal{V}^p_{k\tau}| \geq 1$ **then**
      forall $\rho \in \mathcal{V}^p_{k\tau}$ **do**
        Let $\rho^d_{k\tau} = \tilde{a}^d_{k\tau} - d_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau}$
        Let $\rho^t_{k\tau} = \tilde{a}^t_{k\tau} - d_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau}(\eta_k)^{-1}$
      if *there exists at least one* $\rho \in \mathcal{V}^p_{k\tau}$ *such that* $\rho^d_{k\tau} \geq A^d_{k\rho}$ *and* $\rho^t_{k\tau} \geq A^t_{k\rho}$ **then**
        $EndofRoute_{k\tau} \leftarrow max\{\rho, \rho^d_{k\tau}\}$
      else if $|\mathcal{V}^b_{k\tau}| \geq 1$ **then**
        forall $b \in \mathcal{V}^b_{k\tau}$ **do**
          Let $b^d_{k\tau} = \tilde{a}^d_{k\tau} - d_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)bk\tau}$
          Let $b^t_{k\tau} = \tilde{a}^t_{k\tau} - d_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)bk\tau}(\eta_k)^{-1}$
        if *there exists at least one* $b \in \mathcal{V}^b_{k\tau}$ *such that* $b^d_{k\tau} \geq 0$ *and* $b^t_{k\tau} \geq 0$ **then**
          $EndofRoute_{k\tau} \leftarrow max\{b, b^d_{k\tau}\}$
        else
          There exists no feasible end-of-route assignment for the current route
      else
        There exists no feasible end-of-route assignment for the current route
    else
      Go to step 11

certain model management features in which subjective requirements are dictated by the MLE response selection operators on a temporal basis. The encoding scheme proposed for use in this dissertation is illustrated for a hypothetical problem instance in time stage τ' with the following parameters: $\mathcal{V}^b = \{B_1, B_2\}$, $\mathcal{V}^r = \{a, b, c\}$, $\mathcal{V}^r_{0\tau'} = \{0_a, 0_b, 0_c\}$, $\mathcal{V}^e_{\tau'} = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $\mathcal{V}^p = \{P_1, P_2, P_3, P_4\}$.

In the literature, solutions to a VRP instance are typically encoded as *strings*, which comprise substrings representing routes consisting of a subset of customers scheduled to be visited by a particular vehicle. The order in which customers are entered in such a substring is also the order in which the assigned vehicle visits them along its route. An example of such a solution string for the above hypothetical MLE response selection problem instance is presented in Figure 7.2. In the first route (substring), for instance, MLE resource a is scheduled to first visit VOI 2 from its departure point 0_a, and then VOI 5, after which it is scheduled to relocate to base B_1. In terms of the decision variables of the model formulation of Chapter 4, this part of the solution associated with MLE resource a may be written as $x_{0_a 2a\tau'} = 1$, $x_{25a\tau'} = 1$, $x_{5B_1a\tau'} = 1$, and $x_{ija\tau'} = 0$ otherwise. In addition, $y_{2a\tau'} = 1$, $y_{5a\tau'} = 1$, and $y_{ia\tau'} = 0$ otherwise.

0_a	2	5	B_1	0_b	3	6	7	B_2	0_c	8	4	1	P_4

FIGURE 7.2: *Example of a solution string encoding for the MLE response selection problem.*

The string encoding scheme described above may be simplified by first removing the initial and end-of-route cells, and then inserting dummy cells containing zero elements instead as route delimiters, as shown in Figure 7.3. In this simplified string configuration, the sequence of integers between any two specific zeros therefore represents the order of the VOIs that have to be visited by a specific MLE resource in the set $\mathcal{V}^r$. In the case of idle MLE resources, the respective routes are, of course, left empty. This is represented by means of two consecutive zeros within the string.

2	5	0	3	6	7	0	8	4	1

FIGURE 7.3: *Simplified solution string encoding with route delimiters corresponding to the expanded string encoding of Figure 7.2.*

Although not indispensable, part of configuring a solution string involves accommodating the various complexities associated with the dynamic model constituents of the problem, which renders a solution highly sensitive to being ejected into the infeasible region of decision space whilst employing certain global search operators. This may create an excessive rate of infeasible neighbouring solutions generated during a stochastic solution transformation process, which may severely handicap the progression of the solution search process in the sense of wasting considerable amounts of time finding feasible transformations.

In particular, the use of VOI inclusion sets, as proposed in §5.2.2, suggests that any solution transformation resulting in the removal of one or more VOIs belonging to inclusion sets of their respective routes will generate infeasible neighbouring solutions. Even more extreme, the use of VOI ordered inclusion sets requires the VOI to be reinserted in exactly the right position within a specific route so as to maintain feasibility in that regard. Although not as extreme, the use of VOI exclusion sets, as proposed in §5.2.1, similarly implies that any solution transformation resulting in the insertion of one or more VOIs within a route belonging to the

corresponding exclusion sets will generate infeasible neighbouring solutions. Provided that the solution exploration transformations are performed at random, this problem is expected to be exacerbated as a result of three factors: (1) the number of VOIs belonging to VOI inclusion sets and VOI exclusion sets (where more is worse), (2) the number of scheduled active MLE resources (where fewer is worse), and (3) the probability of conducting an exploration transformation of the current solution (where a higher probability is worse). Furthermore, the combined assignment features presented in §5.3.1 may require certain cells (other than the dummy delimiter cells containing zero elements) to appear multiple times within specific routes and at specific positions within these routes.

One way of alleviating this shortcoming is to remove the VOIs belonging to VOI inclusion sets, VOI exclusion sets and combined assignments from the solution string, then to carry out the solution transformation process with respect to the reduced string, and finally to reinsert these VOIs strategically into feasible positions in such a way that they agree with the definitions implied by these sets[1].

Another way of eliminating this shortcoming is to simply not consider VOIs belonging to such model management sets as candidate vertices to solution exploration transformation procedures. This method should preferably be employed whenever a relatively large number of VOIs is present, as reinserting those (even strategically) every time into a reduced string might be detrimental to the efficient exploitation of solutions mapped closely to or onto the Pareto front.

Returning to the example above, suppose that the operator believes, based on his experience, that it is necessary to fix $\mathcal{I}_{a\tau'} = \{2\}$ and $\mathcal{E}_{b\tau'} = \{4\}$ as part of the MLE response selection operations conducted during the current time stage. Adopting the first alternative above, VOIs belonging to any of these sets may then temporarily be removed from the current string, as shown in Figure 7.4.

5	0	3	6	7	0	8	1

FIGURE 7.4: *Solution string encoding after removing VOIs belonging to inclusion and exclusion sets.*

In the solution methodology proposed in this section, the overall solution transformation process consists of three independent, successive sub-transformations performed on a reduced string. These sub-transformations processes are called *general* sub-transformations, *reinsertion* sub-transformations and *end-of-route assignment* sub-transformations. A general sub-transformation is first performed, during which a random reduced neighbouring string is generated from the current reduced string. In Figure 7.5, an inter-route sub-transformation is performed by removing VOI 3 and VOI 6 from the second substring (route) and reverse-inserting it into the first substring, while VOI 5 is removed from the first substring and inserted into the second substring.

6	3	0	7	5	0	8	1

FIGURE 7.5: *A general sub-transformation performed on the reduced string encoding of Figure 7.4.*

Following this sub-transformation, the VOIs belonging to inclusion and exclusion sets that were

[1]This does not mean that the generated neighbouring solution will always be feasible, as there are other (unrelated) routing feasibility criteria that also have to be considered.

temporarily removed previously are replaced in the resulting reduced neighbouring string according to a reinsertion sub-transformation, as shown in Figure 7.6. This reinsertion procedure may be conducted in several different ways. The simplest method is to reinsert such vertices at randomly chosen positions in feasible routes with respect to the definition of these sets only (the VOIs belonging to ordered inclusion sets would, however, have to be reinserted in specific positions within these routes). Other procedures involve reinserting these vertices in carefully chosen positions within certain feasible routes or sub-routes. For example, with respect to VOI inclusion sets, given a randomly chosen feasible route, one may choose a position within that route that produces the smallest resulting increase in operating costs or in delay time. Moreover, with the use of VOI exclusion sets, one may choose to reinsert the VOI in a route that produces the largest resulting increase in the visitation score.

2	6	4	3	0	7	5	0	8	1

FIGURE 7.6: *A reinsertion sub-transformation performed on the reduced string encoding of Figure 7.5.*

Finally, it is required to schedule every active MLE resource in the string to either a base or a patrol circuit using an end-of-route assignment sub-transformation. Considering the example above again, suppose that the idle MLE resource operator has imposed the following conditions with respect to the post-mission allocation of MLE resources for the current time stage (see §5.2.3): $\mathcal{V}^{b}_{a\tau'} = \{B_2\}$, $\mathcal{V}^{p}_{a\tau'} = \{P_1\}$, $\mathcal{V}^{b}_{b\tau'} = \{B_1, B_2\}$, $\mathcal{V}^{p}_{b\tau'} = \emptyset$, $\mathcal{V}^{b}_{c\tau'} = \{B_1\}$ and $\mathcal{V}^{p}_{c\tau'} = \{P_3, P_4\}$. Furthermore, based on the nature of the reduced solution string in Figure 7.6, assume that B_1 is spatially closer to VOI 5 than B_2 is, that P_3 is spatially closer to VOI 1 than P_4 is, and that MLE resource a is expected to have low autonomy levels after completing its current mission, which is to visit VOIs 2, 6, 4 and 3 (in that order). After removing the dummy cells and inserting bases or patrol circuits from the sets presented above at the end of their respective routes according to Algorithm 7.1, the complete neighbouring string corresponding to the original string in Figure 7.2 is shown in Figure 7.7.

0_a	2	6	4	3	B_2	0_b	7	5	B_1	0_c	4	1	P_3

FIGURE 7.7: *An end-of-route assignment sub-transformation performed on the reduced string encoding of Figure 7.6.*

In summary, a flowchart of this proposed overall solution transformation procedure is presented in Figure 7.8. This procedure may easily be incorporated in an optimisation search engine, especially since the three proposed sub-transformations are conducted independently from one another.

7.4 Pareto front approximation

In a multiobjective combinatorial optimisation problem there exists a countable number of solutions on the Pareto front. The aim of any solution search technique is to approximate these Pareto-optimal solutions as closely as possible. In general, however, a close approximation of the entire Pareto front during a single run of such a solution search technique is almost impossible for complex multiobjective problems, especially when faced with a strict *computational budget* (*i.e.* if

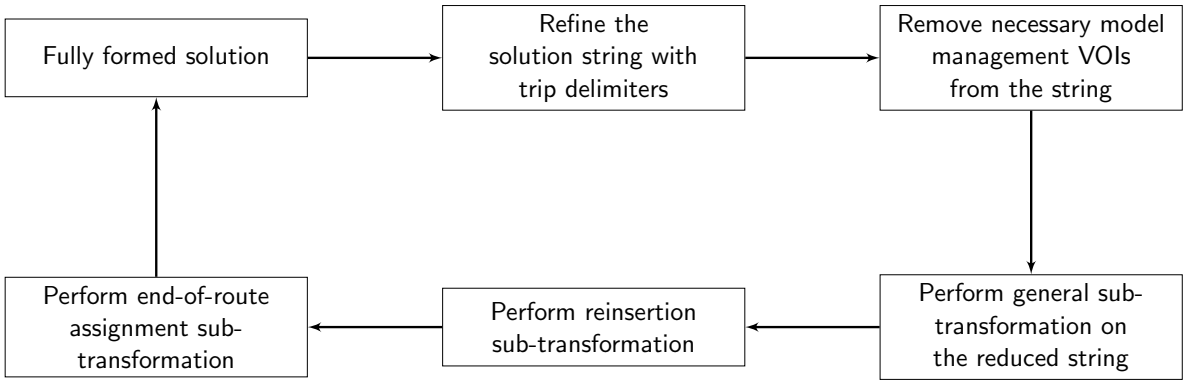


FIGURE 7.8: *Flow of events in the proposed overall solution transformation procedure.*

the computational resources available for solving the problem, such as the search time, processing power or storage capacity, are tightly constrained). Moreover, finding a good approximation of even a portion of the Pareto front is difficult when the structure of the underlying domain space is not known or is very disjoint [44].

According to Zitzler *et al.* [167], a multiobjective search technique should always aim to achieve three conflicting goals with respect to the nature of the resulting non-dominated front. First, the non-dominated front should ideally be as close as possible to the true Pareto front. The non-dominated set of solutions should, in fact, preferably be a subset of the true Pareto-optimal set if possible. Secondly, solutions in the non-dominated set should ideally be uniformly distributed and diverse with respect to their objective function values along the Pareto front so as to provide the decision maker with the full picture of trade-off decisions. Lastly, the non-dominated front should ideally capture the entire range of values along the true Pareto front.

Here, the first goal is best served by focusing the search on a particular region of the true Pareto front — a process known in the literature as *intensification* (or *exploitation*). On the other hand, the second goal requires the search technique to investigate solutions that are uniformly distributed along the true Pareto front — a process known in the literature as *diversification* (or *exploration*). Finally, the third goal aims to extend the non-dominated front at both ends of the true Pareto front in order to explore new, extreme solutions.

In the next two sections, two familiar solution search methodologies which may be incorporated into the optimisation methodology component of the MLE response selection DSS of this dissertation are presented, namely the archived multiobjective simulated annealing algorithm (mentioned in §2.3.3) and the NSGA-II (mentioned in §2.4.2). These solution search techniques have been documented extensively in the literature as being capable of achieving the above three goals for a wide variety of problems, provided that their algorithmic configurations (parameter values) are chosen carefully. Furthermore, a hybrid approach combining these two algorithms and two multi-start simulated annealing algorithms adapted from the (single-start) algorithm in [131] are also developed.

7.5 Multiobjective simulated annealing methodology

In §2.3.3, a brief review of key multiobjective simulated annealing techniques from the literature was conducted and it was mentioned in §2.6.5 that certain of these techniques may be adapted for solving multiobjective VRPs when tuned appropriately (in terms of parameter value choices).

In this section, the innovative multiobjective simulated annealing algorithm proposed for this purpose by Smith *et al.* [130, 131] is described in some detail, following which certain simulated annealing solution transformation methods which comply with the unique nature of the MLE response selection problem are proposed.

7.5.1 The notion of archiving

Because the method of simulated annealing only generates a single solution during each iteration of the search process, an external set, known as the *archive*, is employed to record all non-dominated solutions uncovered during the course of the search process. All solutions generated during the course of the search are candidates for archiving, and are each tested for dominance in respect of every solution in the archive. The archiving process is illustrated in Figure 7.9 for a bi-objective minimisation problem with objective functions f_1 and f_2.

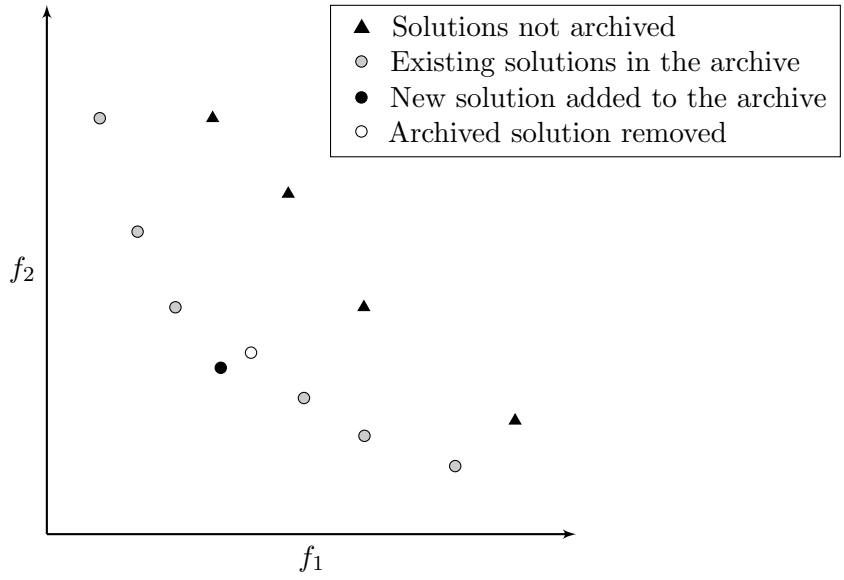


FIGURE 7.9: *The archiving process for a bi-objective minimisation problem.*

If the true Pareto front $\mathcal{P}$ were available *a priori*, it would be possible to express the energy (*i.e.* the performance of a solution in objective space) of a solution $\boldsymbol{x}$ as some measure of the portion of the front that dominates $\boldsymbol{x}$. To this effect, let $\mathcal{P}_{\boldsymbol{x}}$ be the portion of $\mathcal{P}$ that dominates $\boldsymbol{x}$ (*i.e.* $\mathcal{P}_{\boldsymbol{x}} = \{\boldsymbol{y} \in \mathcal{P} \,|\, \boldsymbol{y} \succ \boldsymbol{x}\}$), and define the energy of $\boldsymbol{x}$ as $E(\boldsymbol{x}) = \mathcal{M}(\mathcal{P}_{\boldsymbol{x}})$, where $\mathcal{M}$ is a measurable function defined on $\mathcal{P}$. If $\mathcal{P}$ is continuous, $\mathcal{M}$ may be configured as the Lebesgue measure; otherwise $\mathcal{M}$ may simply be configured as the cardinality of $\mathcal{P}_{\boldsymbol{x}}$ (*i.e.* the number of solutions in $\mathcal{P}$ that dominate $\boldsymbol{x}$). Of course, $E(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathcal{P}$.

As the true Pareto front is typically unavailable during the course of the search, however, the energy of a solution is instead measured based on the current estimate of the true Pareto front, which corresponds to the set of non-dominated solutions uncovered thus far during the search (that is, the solutions in the archive). The energy difference between two solutions is measured as the quotient of the difference in their respective energies and the size of the archive. According to Smith *et al.* [131], a simulated annealer that uses this energy measure encourages both convergence to and coverage of the Pareto front in logarithmic computational time.

7.5.2 Algorithm outline

Suppose that the set of non-dominated solutions uncovered up to any point during the solution search process is captured in an archive $\mathcal{A}$. The algorithm is initialised with a random feasible initial solution, which is initially placed in $\mathcal{A}$. During any iteration, let $\tilde{\mathcal{A}}$ be the set $\mathcal{A} \cup \{\boldsymbol{x}\} \cup \{\boldsymbol{x}'\}$, where $\boldsymbol{x}'$ is a neighbouring solution generated by performing a transformation of the current solution $\boldsymbol{x}$, and define $\tilde{\mathcal{A}}_{\boldsymbol{x}} = \{\boldsymbol{y} \in \tilde{\mathcal{A}} \,|\, \boldsymbol{y} \succ \boldsymbol{x}\}$.

The estimated energy difference between the solutions $\boldsymbol{x}$ and $\boldsymbol{x}'$ is then calculated as

$$\Delta_E(\boldsymbol{x}', \boldsymbol{x}) = \frac{|\tilde{\mathcal{A}}_{\boldsymbol{x}'}| - |\tilde{\mathcal{A}}_{\boldsymbol{x}}|}{|\tilde{\mathcal{A}}|},$$

where division by $|\tilde{\mathcal{A}}|$ provides robustness against fluctuations in the number of solutions in the archive. The reason for including both the current and neighbouring solutions in $\tilde{\mathcal{A}}$ is motivated by the idea that $\Delta_E(\boldsymbol{x}', \boldsymbol{x}) < 0$ whenever $\boldsymbol{x}' \succ \boldsymbol{x}$. Besides its simplicity and efficiency in promoting the storage of non-dominated solutions uncovered during the search process, another benefit of this energy measure is that it encourages exploration along the non-dominated front, regardless of the portion of the true Pareto front dominating a solution. This principle is demonstrated in Figure 7.10 for a bi-objective minimisation problem with objective functions f_1 and f_2. Although it appears as if $\mathcal{M}(\mathcal{P}_{\boldsymbol{x}'}) > \mathcal{M}(\mathcal{P}_{\boldsymbol{x}})$ in the figure, it can be seen that $|\tilde{\mathcal{A}}_{\boldsymbol{x}'}| = 1 < 3 = |\tilde{\mathcal{A}}_{\boldsymbol{x}}|$, hence moving the search closer to a large unexplored region of the non-dominated front when transitioning from $\boldsymbol{x}$ to $\boldsymbol{x}'$.

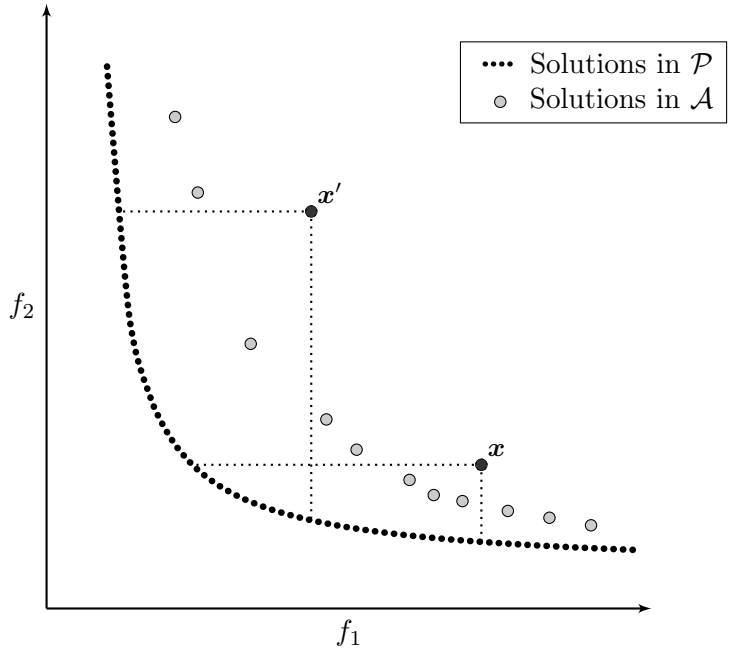


FIGURE 7.10: *Energy measure of current and neighbouring solutions for a bi-objective minimisation problem with objective functions f_1 and f_2.*

The acceptance probability function of a neighbouring solution $\boldsymbol{x}'$ of the current solution (similar to the Metropolis acceptance rule in standard simulated annealing algorithms) is taken as

$$P(\boldsymbol{x}') = \min\left\{1, \exp\left(\frac{-\Delta_E(\boldsymbol{x}', \boldsymbol{x})}{T}\right)\right\},$$

where T is the current temperature. A neighbouring solution that is dominated by *fewer* solutions in $\mathcal{A}$ therefore has a *lower* energy and is consequently automatically accepted as the next current solution as it is considered, by definition, to embody an improving move. On the other hand, if there is a large positive difference between the energies of the neighbouring and current solutions, and the temperature is low, then the move has a low probability of being accepted. This acceptance probability function therefore does not depend upon pre-determined objective function weights and is not affected by rescaling of the objectives, which results in relatively low computational complexity. If the move is accepted, then the neighbouring solution is taken as the new current solution and the archive is updated accordingly, as described in the previous section.

A flowchart of the dominance-based method of archived multiobjective simulated annealing is presented in Figure 7.11 for a minimisation problem. In addition, a pseudo-code description of the basic steps of the algorithm is given in Algorithm 7.2. Without loss of generality, the cooling schedule in this pseudo-code is represented by a sequence of temperatures $T_1, \ldots, T_C$, where $T_c > T_{c+1}$ for all $c \in (1, \ldots, C-1)$, with associated cooling epoch durations $L_1, \ldots, L_C$ (*i.e.* the number of iterations performed at each corresponding temperature level). The stopping criterion of the algorithm is therefore based on a maximum number of iterations $I_{max} = \sum_{c=1}^{C} L_c$.

Algorithm 7.2: Dominance-based method of archived multiobjective simulated annealing

Input : A multiobjective minimisation problem with sets of variables, constraints and objective functions, a cooling schedule $(T_1, \ldots, T_C)$ with corresponding epoch durations $(L_1, \ldots, L_C)$, and a maximum number of iterations I_{max}.

Output: A non-dominated set of solutions $\tilde{\mathcal{P}}$ in decision space and the corresponding set of performance vectors $\tilde{\mathcal{F}}$ in objective space.

Generate initial feasible solution $\boldsymbol{x}$
Initialise archive $\mathcal{A} = \{\boldsymbol{x}\}$
Initialise cooling schedule epoch $c \leftarrow 1$ with $L_0 = 0$
Initialise iteration counter $t \leftarrow 1$
while $t \leq I_{max}$ **do**
  while $t \leq \sum_{a=1}^{c} L_{a-1} + L_c$ **do**
    Generate neighbour solution $\boldsymbol{x}'$ from current solution $\boldsymbol{x}$
    while $\boldsymbol{x}'$ *is infeasible* **do**
      Generate neighbour solution $\boldsymbol{x}'$ from current solution $\boldsymbol{x}$
    Assess $\Delta_E(\boldsymbol{x}', \boldsymbol{x})$
    Generate random number $r \in (0, 1)$
    if $r < \min\left\{1, \exp\left(\frac{-\Delta_E(\boldsymbol{x}', \boldsymbol{x})}{T_c}\right)\right\}$ **then**
      $\boldsymbol{x} \leftarrow \boldsymbol{x}'$
      if $|\mathcal{A}|_{\boldsymbol{x}} = 0$ **then**
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{\boldsymbol{x}\}$
        forall $\boldsymbol{y} \in \mathcal{A}$ **do**
          if $\boldsymbol{x} \succ \boldsymbol{y}$ **then**
            $\mathcal{A} \leftarrow \mathcal{A} \backslash \{\boldsymbol{y}\}$
    $t \leftarrow t + 1$
  $c \leftarrow c + 1$
$\mathcal{P} \approx \tilde{\mathcal{P}} = \mathcal{A}$

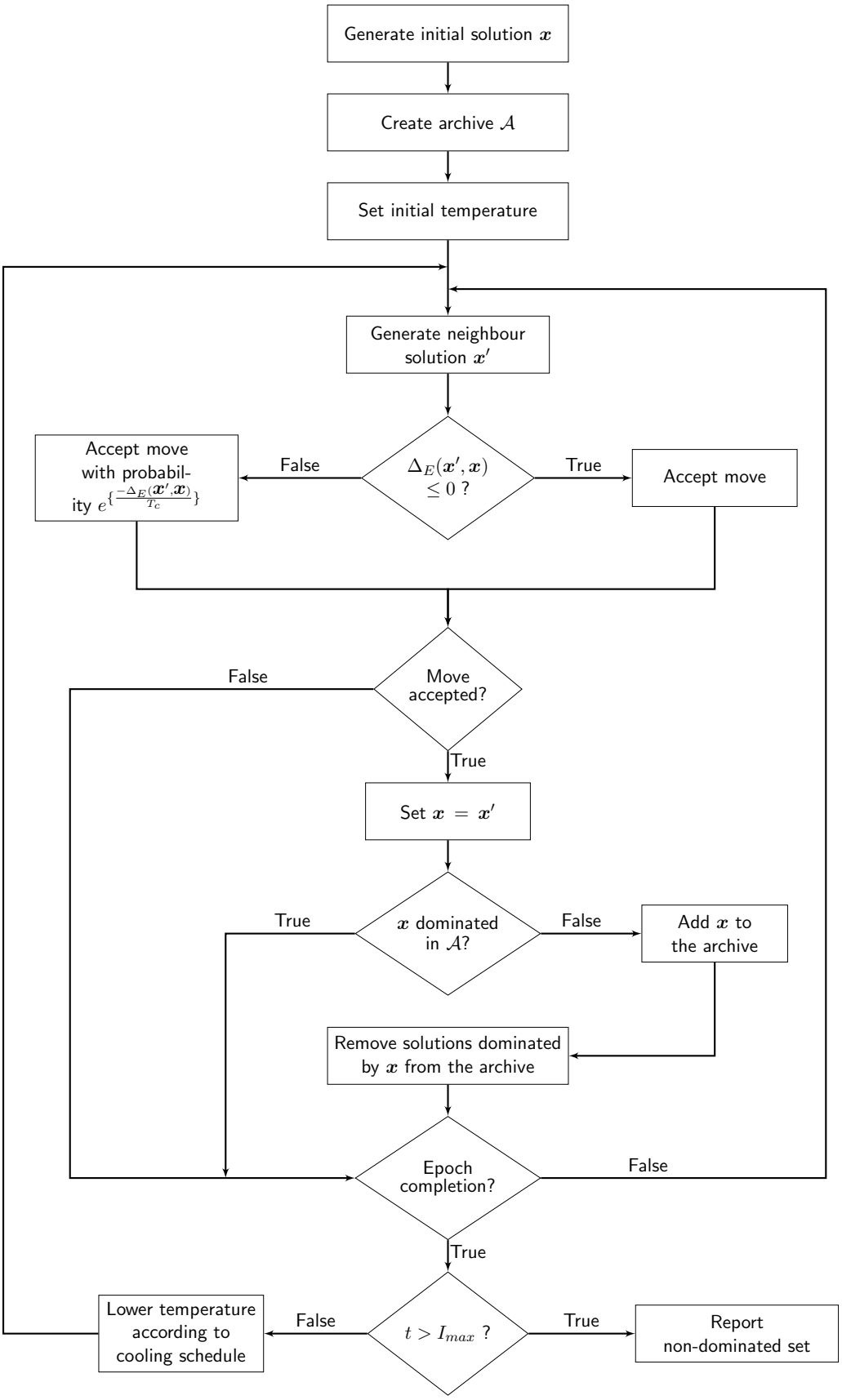


FIGURE 7.11: *Flow of events in the dominance-based, archived multiobjective simulated annealing algorithm of Smith* et al. *[131].*

7.5.3 General solution sub-transformation procedures

Two general types of solution transformation methods are typically employed in the simulated annealing literature for generating neighbouring solutions in a VRP, namely *intra-route* transformations and *inter-route* transformations [156]. An intra-route transformation technique changes a route within the same solution string by altering one or more partial strings within that route in an attempt to improve it. An inter-route transformation, on the other hand, exchanges or relocates one or more partial strings between any routes of a VRP solution string. These transformations are used to encourage exploitation and diversification of the search, respectively.

Researchers are often required to improvise their own solution transformation techniques in order to better accommodate certain types of VRPs and/or try to improve the search process in general. In [156], for example, three inter-route improvement methods are proposed. These are the *string exchange method*, where two partial strings from two different routes are exchanged with one another at specific locations within the routes, the *string relocation method*, where a partial string containing one or more entries from one route is inserted into a specific location within another route, and the *string mix method*, which attempts to relocate or exchange partial strings.

More generic methods of transformation are the *one-to-one exchange*, *delete-and-insert* and *partial string reversal* [80, 140, 156], which are applicable to both intra-route and inter-route transformations. These techniques also allow for the zero delimiters in a string to be moved around, so as to change the nature and size of the routes pre-assigned to the vehicles. The first two methods are similar to the string exchange and string relocation methods, respectively, as described in the previous paragraph. The partial reversal method consists of temporarily removing part of the string, reversing the direction of that partial string, and either placing it back where it was originally removed or inserting it somewhere else in the string.

In addition, however, the VRP customer visitation profits characteristic associated with the MLE response selection problem requires transformations that alter the number of VOIs scheduled to be intercepted by adding or removing partial strings to or from the current solution. Such transformations are defined in this dissertation as *inter-state* transformations, where the *state* of a solution is defined as the number of VOIs visited in a solution. More specifically, a *string expansion* transformation is defined as one that increases the number of VOIs in the current solution string by inserting one or more partial strings containing currently unvisited VOIs stored from an external set into it. A *string diminution* transformation is similarly defined as one that decreases the number of VOIs in the current solution string by removing one or more partial strings from it.

Examples of some of the above solution transformation procedures for VRP solution strings are illustrated in Figure 7.12. For the sake of clarity, note from §7.3 that these actually constitute general sub-transformation procedures. Here, Figures 7.12 (a) and (b) represent intra-route transformations, while Figures 7.12 (c) and (d) represent inter-route transformations. Figures 7.12 (e) and (f) finally represent inter-state transformations.

At every iteration, it is therefore proposed that a type of solution transformation move be selected according to a pre-configured probability distribution, where the parameter $P_{WR} \in [0, 1]$ is the probability of conducting an intra-route transformation, while the parameter $P_{BR} \in [0, 1]$ is the probability of conducting an inter-route transformation and the parameter $P_S \in [0, 1]$ is the probability of conducting an inter-state transformation. Of course, $P_{WR} + P_{BR} + P_S = 1$. In addition, there may be several sub-types of moves within each of these transformation classes, which are also triggered according to a certain probability distribution. For example, the user may wish to conduct a delete-and-insert or a partial reversal move according to certain

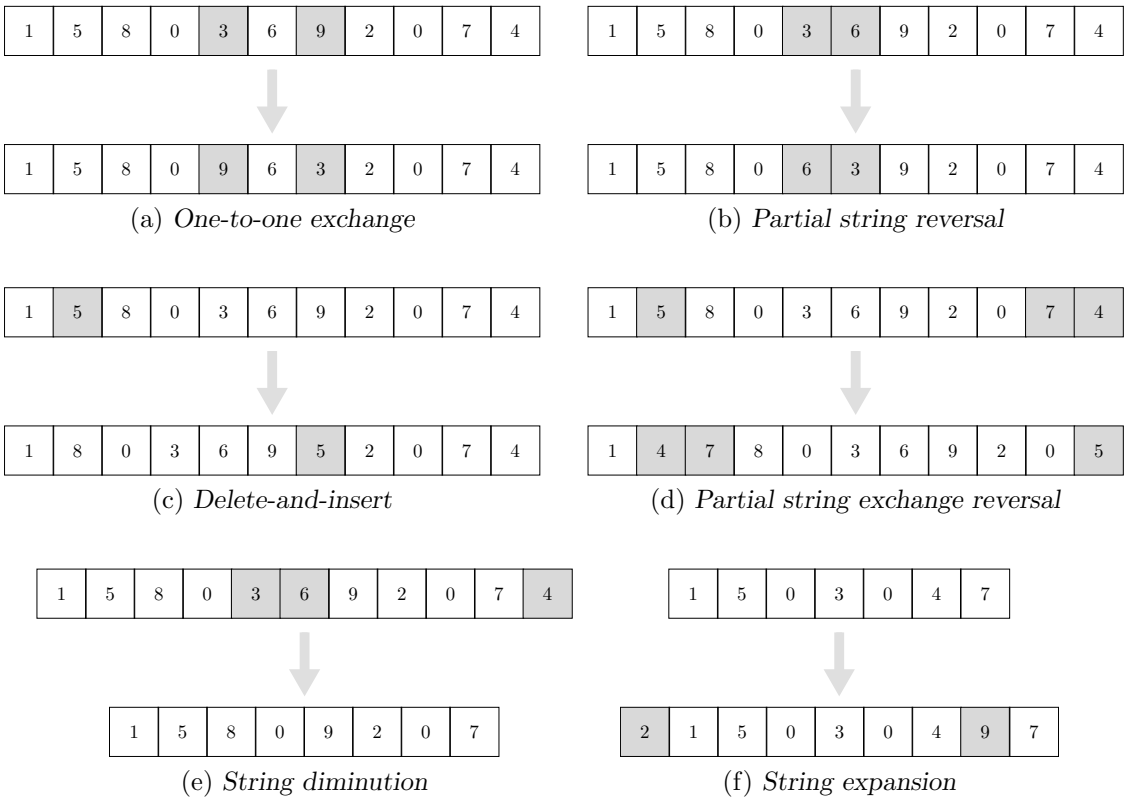


FIGURE 7.12: *Illustration of solution transformation procedures.*

probabilities whenever an intra-route transformation is triggered. The choice of transformation moves and associated probabilities may typically differ from one user or situation to another.

7.5.4 Parametric configurations for inter-state transformations

The most difficult solution transformation-related process in solving the MLE response selection problem evolves around efficiently transitioning the search through the state subsets of the decision space with the use of inter-state solution transformations. In particular, it is important to acknowledge that the size of the subsets of the decision space associated with the states of solutions increases exponentially with the sizes of the states. In other words, considerably more solutions reside in decision subspaces associated with higher states. More importantly, however, it is also critical to acknowledge that this is not necessarily true in exclusive respect of the feasible decision space.

Configuring such a process is particularly challenging given that no research in this respect could be uncovered in the literature. Nevertheless, employing the fundamental structure of discrete probability spaces, three functional parametric configurations are proposed next, namely (1) an *exponential growth distribution* approach, (2) a *binomial distribution* approach and (3) a *states clustering* approach. Overall, the proposed configurations are particularly useful considering that they are generic enough to be applicable to any MLE response selection problem instance, and are not constrained by the size of the set of VOIs or the size of the set of MLE resources. Additionally, these configurations are expected to be able to provide a full range of transformations across all states, while simultaneously promoting exploitation and exploration of the search space.

To be more specific, the first configuration approach above is best suited for solving MLE response selection problems in which the feasible decision space grows exponentially with the size of the states, and in which the number of states is not large. The second approach, on the other hand, is best suited for solving problems in which the highest states do not monopolise a significantly high proportion of the entire feasible decision space or, analogously, if these states contain a significant proportion of infeasible solutions within their respective domain subspaces (which may, in turn, waste valuable computational budget by generating too many infeasible solutions during the search process). Lastly, the third approach is believed to generally be most effective (in comparison to the other two approaches) when a large number of states is present, regardless of the nature of the underlying state subspaces.

The exponential growth distribution approach

Provided that an inter-state transformation has been triggered, it is required to establish a range of probabilities associated with entering any other state given that the string is currently in a certain state. Let $\tilde{n}_\tau \leq n_\tau$ represent the number of VOIs in the reduced string of the current solution. The proposed probability of performing a transformation that moves the search to a solution in state $s \in \{1, \ldots, \tilde{n}_\tau\}$, provided that it currently is in state $s' \in \{1, \ldots, \tilde{n}_\tau\}$ with $s \neq s'$, is given by

$$P_{s's} = \begin{cases} P_S \left(1 + \sum_{s=2}^{\tilde{n}_\tau} \alpha s^\beta - \alpha s'^\beta\right)^{-1}, & \text{if } s = 1 \\ P_S \alpha s^\beta \left(\sum_{s=2}^{\tilde{n}_\tau} \alpha s^\beta\right)^{-1}, & \text{if } s' = 1 \\ P_S \alpha s^\beta \left(1 + \sum_{s=2}^{\tilde{n}_\tau} \alpha s^\beta - \alpha s'^\beta\right)^{-1}, & \text{otherwise,} \end{cases} \tag{7.6}$$

where $\alpha, \beta \in \Re^+$ are respectively linear and exponential user-defined parameters tuned according to preferences in respect of the expected proportion of the search spent amongst the states. Here, larger values of α and β increase the expected proportion of the search process spent searching in subspaces containing solutions corresponding to larger states, with β being more sensitive to incremental changes as a result of its exponential nature. This approach therefore requires two-user defined parameters. The combined use of these two parameters results in a greater variety of inter-state configurations, and therefore in a greater degree of flexibility, in comparison to the other two approaches. Moreover, using this approach, it is easy for the user to simply choose one fixed configuration of α and β, and use it over multiple time stages (*i.e.* the user is most certainly not required to redefine these parameters at the start of every search process).

It is possible to demonstrate that

$$\sum_{\substack{s \in \{1, \ldots, \tilde{n}_\tau\} \\ s \neq s'}} \frac{P_{s's}}{P_S} = 1, \qquad s' \in \{1, \ldots, \tilde{n}_\tau\},$$

which is a necessary identity for a feasible configuration of this solution transformation process. Based on (7.6), the probability of moving into any other state provided that the solution is currently in a certain state may be modelled as an *ergodic Markov chain*, as illustrated in Figure 7.13, with transition probability matrix

$$\begin{array}{c} \\ s \end{array} \begin{array}{c} \begin{array}{cccc} & & s' & \end{array} \\ \begin{bmatrix} 0 & P_{12} & \ldots & P_{1\tilde{n}_\tau} \\ P_{21} & 0 & \ldots & P_{2\tilde{n}_\tau} \\ \vdots & \vdots & \vdots & \vdots \\ P_{\tilde{n}_\tau 1} & P_{\tilde{n}_\tau 2} & \ldots & 0 \end{bmatrix} \end{array}$$

and steady-state probabilities

$$\left[\frac{1}{1+\sum_{s=2}^{\tilde{n}_\tau}\alpha s^\beta} \quad \frac{\alpha(2^\beta)}{1+\sum_{s=2}^{\tilde{n}_\tau}\alpha s^\beta} \quad \cdots \quad \frac{\alpha(\tilde{n}_\tau^\beta)}{1+\sum_{s=2}^{\tilde{n}_\tau}\alpha s^\beta}\right].$$

According to the *law of large numbers* [52], the expected proportion of the search spent in each state therefore approaches that of the entries in the vector above as the number of algorithmic iterations approaches infinity. Again, it is possible to demonstrate that the sum of these entries add up to 1. These steady-state probabilities may, however, not accurately reflect the desired search repartition if the number of iterations performed by the algorithm is too small and the number of states is too large.

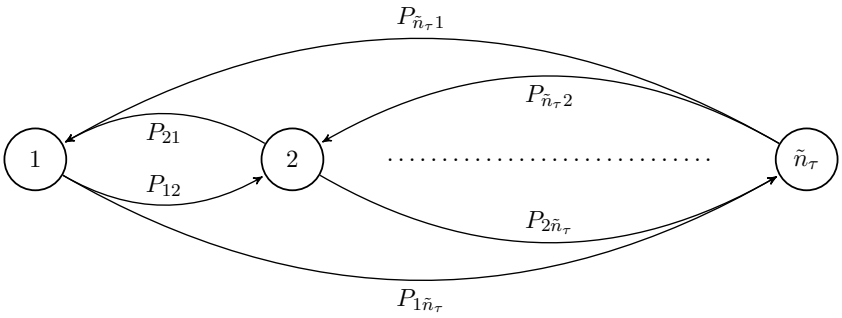


FIGURE 7.13: *Markov chain of inter-state transition probabilities.*

The binomial distribution approach

A random variable $U \sim (\tilde{n}, p)$ is governed by a *binomial probability distribution* based on $\tilde{n}$ trials with success probability p if

$$P(U=u) = \frac{\tilde{n}!}{u!(\tilde{n}-u)!}p^u(1-p)^{\tilde{n}-u}, \qquad u \in \{0,1,\ldots,\tilde{n}\}.$$

Here, the distribution parameter $\tilde{n}$ once again represents the maximum number of VOIs that may be visited in a reduced solution string, while the user-defined parameter p is responsible for steering the expected proportion of the search spent in every state. This approach therefore only requires one user-defined parameter, where the search process is expected to spend more time investigating higher-state solutions whenever a higher success probability parameter is employed. Furthermore, the probability of performing a transformation that moves the search to a solution in state $u \in \{0,1,\ldots,\tilde{n}\}$, provided that it currently is in state $u' \in \{0,1,\ldots,\tilde{n}\}$ with $u \neq u'$, is given by

$$P_{u'u} = P_S\left(\frac{P(U=u)}{1-P(U=u')}\right),$$

where

$$\sum_{\substack{u\in\{0,1,\ldots,\tilde{n}\}\\ u\neq u'}} \frac{P_{u'u}}{P_S} = 1, \qquad u' \in \{0,1,\ldots,\tilde{n}\}.$$

Because state 0 contains only one solution, which may easily be evaluated *a priori*, it is noted that it would be impractical to allocate a computational budget to access this state. If $P(U=0)$ is very small, this issue is trivial. If this probability is not insignificant, however (such as in

scenarios where the maximum number of states $\tilde{n}$ is small and the steering parameter p is not set too high), then an alternative inter-state transformation approach should be employed, or the distribution should be recalibrated accordingly so as to prevent access to state 0 within the underlying Markov chain.

Similarly to the first approach, the parameter p need not be redefined by the user at the start of every run of the search process. It may, for example, be pre-set as a function of the number of VOIs and MLE resources available at the beginning of a problem instance. Moreover, the sum of all ideal quantities of VOI interceptions ($\sum_{s\in\mathcal{Z}} N_{s\tau}$), as provided by the various decision entities (see §4.3), may also serve as a guideline as to the choice of an appropriate value for this parameter.

The states clustering approach

Due to the nature of the binomial distribution, states that are located far away from the mean are unlikely to be visited during the search process[2], and this is more pronounced when a large number of states is present. Consequently, the decision maker may prefer to adopt a *states clustering* approach as a remedying alternative, in which groups of states are regarded as the outcomes of a certain binomial (or similarly applicable) distribution. Then, upon selection of a certain outcome, any state within the cluster represented by this outcome is entered with a certain probability, based on the number of states within the cluster as well as the cardinalities of these states. In this configuration, states which are far away from the mean are therefore allowed to be visited more frequently during the search process than in the previous approach, while states which are close to the mean value are not favoured as intensively. The states clustering approach may also be merged with the structure of the exponential growth approach if necessary.

7.5.5 Strategic exploration transformation procedures

Given the very limited computational time available to generate a high-quality set of non-dominated solutions at the start of a problem instance, it may be beneficial to minimise the computational budget waste associated with an excessive proportion of infeasible solutions being generated during the course of a search process for scenarios in which the set of all feasible solutions is relatively small compared to the size of the entire search space. This is the case whenever high-state solutions are considered for inter-route or string expansion transformations.

As in numerous other uncapacitated VRPs in the literature, and as demonstrated in Chapter 6, it is anticipated that over-extended visitation routes are often detrimental to the overall performance of MLE response selection operations due to extended delay times, increased travel costs and higher risk of route failures. Moreover, during the course of a solution search process, it is believed that a considerable portion of solutions are primarily constrained by distance and time autonomy restrictions.

Although perfectly capable of conducting the necessary exploration and exploitation requirements during the course of a search process, the various solution transformations procedures introduced in §7.5.3 have, so far, only been assumed to be performed at random on a solution string. It is therefore alternatively suggested that exploration transformation procedures be

[2]The expected value of a random variable governed by a binomial probability distribution is $\tilde{n}p$, and its variance is $\tilde{n}p(1-p)$.

conducted *strategically* contrary to *randomly* in respect of spatial proximity assignments, particularly in scenarios where VOIs are distributed fairly uniformly over a vast maritime jurisdiction area and/or the MLE resources in place carry small autonomy capacities relative to the size of this jurisdiction area[3]. Two basic examples of strategic solution exploration transformation procedures which may be employed in the context of MLE response selection decision making are briefly presented next. Each of these methods require only one user-defined parameter.

In the first method, a VOI is only allowed to be inserted into or removed from a certain visitation route *via* an exploration move if the current location of the MLE resource assigned to this route is within a pre-defined radius (Euclidean distance) from the current location of the VOI. An example of this exploration filtering procedure is illustrated in Figure 7.14(a) for three MLE resources $\{a, b, c\}$ and four VOIs $\{1, 2, 3, 4\}$, where the radius r_ϵ is fixed for all VOIs, regardless of the nature of their respective anticipated velocity vectors. Here, for instance, VOI 1 may only be inserted into or removed from the visitation routes assigned to MLE resources a and b.

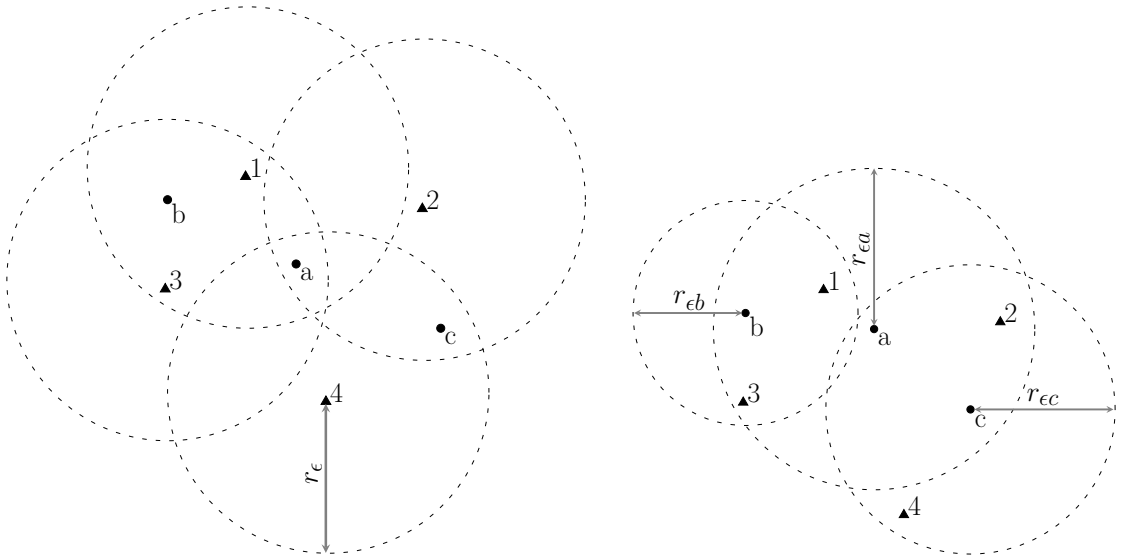
The second method involves the use of radii weighted as a function of the current autonomy levels of the active MLE resources as a means to filter the subset of candidate VOIs that may be inserted into or removed from the visitation route assigned to this MLE resource. An example of this exploration filtering procedure is illustrated in Figure 7.14(b). Here, for instance, VOI 2 may only be inserted into or removed from the visitation routes assigned to MLE resources a and c. Furthermore, assuming that only distance autonomy thresholds are considered, it may be seen that MLE resource a has the highest autonomy level at the beginning of time stage τ, while MLE resource b has the lowest. In other words, $a^d_{a\tau} > a^d_{c\tau} > a^d_{b\tau}$ if and only if $r_{\epsilon a} > r_{\epsilon c} > r_{\epsilon b}$.

Ideally, varying combinations of these two methods may be configured by the user in order to achieve more effective exploration of the search space overall. It is, however, acknowledged that configuring the radii too small may, on the one hand, exclude accessibility to key domain subspaces and, therefore, prevent certain regions of the Pareto front to be reached (similarly to the potentially detrimental consequences of adopting a decentralised MLE response selection decision making paradigm). On the other hand, configuring the radii too large may render this quest for strategic exploration redundant.

7.6 Multiobjective genetic algorithm methodology

Genetic algorithms typically consist of various building blocks, which may be configured in a vast combination of alternatives. Examples of such building blocks include the data structure used to encode population individuals, the procedure of creating an initial population, the evolutionary operators used to generate offspring populations and the termination condition used as a stopping mechanism. Experimenting with and, consequently, identifying the "best" combination of these building blocks for solving multiobjective VRPs of varying problem instance complexities may therefore be a difficult and time consuming task. More importantly, in the MLE response selection problem, where finding a good set of non-dominated solutions within a limited amount of time is deemed more important than finding a Pareto subset of solutions after a relatively long computational time. Furthermore, where the complexity of the problem is generally very high and/or varies from one problem instance to another, it is deemed more important to configure building blocks that will work well for solving many types of problem instances. Alternatively, it is deemed important to develop and implement classes of building

[3]Note that the end-of-route assignment methodology described in §7.2 is another example of spatial proximity strategic solution exploration, where the post-mission allocation of MLE resources to *nearby* bases and patrol circuits are favoured over more *distant* ones.



(a) *Visitation route candidate allocations with fixed VOI radii.*

(b) *Visitation route candidate allocations with weighted MLE resource radii.*

FIGURE 7.14: *Illustrations of strategic exploration transformation procedures with respect to spatial proximity assignments.*

blocks into the DSS that may be used for specific classes of problem instances, rather than having to identify the best combination of building blocks at the start of a problem instance prior to launching the solution search process.

In §2.4.2, a review was presented of key multiobjective genetic algorithm techniques from the literature and it was mentioned in §2.6.5 that certain of these techniques may be used to solve complex multiobjective VRPs. In this section, the working of the non-dominated sorting genetic algorithm II (NSGA-II), proposed by Deb *et al.* [41], is described in some detail and ideas are proposed for generating an initial population and designing unique genetic operators adapted to accommodate the nature of the MLE response selection problem.

7.6.1 Algorithm outline

In this population-based NSGA-II, an initial population of candidate solutions of size N is first randomly generated. These solutions are then ranked and sorted into sets, called *non-dominated fronts*, using the so-called *Fast Non-dominated Sorting Algorithm* (FNSA) [41] as a means of assessing the fitness levels of individuals[4]. The FNSA works as follows. For each individual $\boldsymbol{p}$, the *domination count* $n_{\boldsymbol{p}}^c$ is defined as the number of solutions in the population dominating $\boldsymbol{p}$, while $\mathcal{D}\boldsymbol{p}$ is defined as the set of solutions dominated by $\boldsymbol{p}$. All solutions with a dominance count value $n_{\boldsymbol{p}}^c = 0$ are then placed in a set $\mathcal{F}_1$ called the *first non-dominated front*. Next, for each solution $\boldsymbol{p} \in \mathcal{F}_1$, each individual $\boldsymbol{q} \in \mathcal{D}\boldsymbol{p}$ is visited and its domination count $n_{\boldsymbol{q}}^c$ is reduced by one, thus disregarding the dominance effect of solution $\boldsymbol{p}$ on solution $\boldsymbol{q}$. The solutions with an updated dominance count of 0 are then placed in the second non-dominated front, $\mathcal{F}_2$. The same process is repeated to identify members of a third non-dominated front, and so forth, until all individuals have been placed in some non-dominated front. An example illustrating the notion of domination fronts for a bi-objective minimisation problem with $N = 11$ and objective functions f_1 and f_2

[4]The terms "individuals" and "solutions" are used interchangeably throughout this section.

is shown in Figure 7.15. The non-domination rank $\boldsymbol{p}^{rank} \in \mathbb{N}$ of individual $\boldsymbol{p}$ is then simply defined as the cardinal number of its non-dominated front. A pseudo-code description of this non-dominated sorting approach is presented in Algorithm 7.3. The worst-case computational complexity of this sub-process is $\mathcal{O}(MN^2)$, where M is the number of objective functions. In contrast, the complexity of the non-dominated sorting mechanism used in the original NSGA-I is $O(MN^3)$ [134].

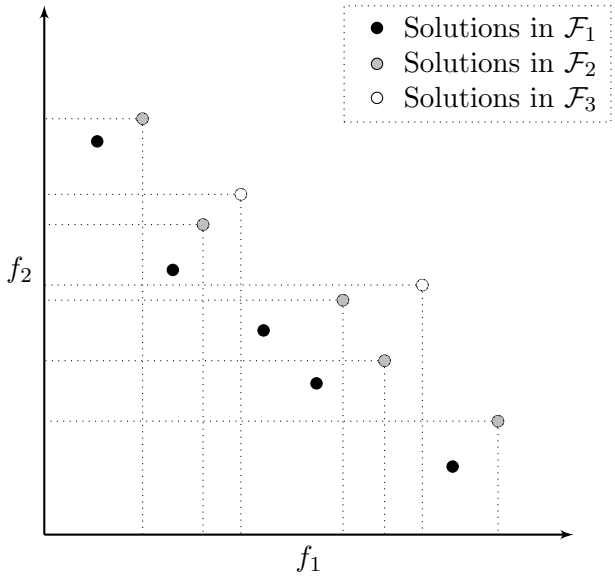


FIGURE 7.15: *Illustration of domination fronts in the NSGA-II for a bi-objective minimisation problem.*

Relatively fast convergence to the Pareto-optimal set (*i.e.* elitism preservation) is one benefit of employing the FNSA described above. However, it is also required to maintain a good spread of solutions within the search space at every generation (*i.e.* maintain diversity preservation), as described in §7.4. In order to achieve this, the algorithm incorporates a simple procedure taking the density estimation of individuals into consideration (that is, an estimate of the density of solutions surrounding an individual in a particular non-dominated front in objective space). The density estimation of an individual $\boldsymbol{p}^{dist}$, called its *crowding distance*, is calculated by taking the average distance of two solutions on either side of that individual along every objective. The solutions are ranked from lowest to highest with respect to each objective function. The boundary solutions (*i.e.* the solutions with the highest and lowest objective function scores) are assigned an infinite crowding distance, while an intermediate solution is assigned a crowding distance that is measured indirectly proportionally to the proximity of its two neighbours alongside each objective function.

The purpose of the crowding distance operator is therefore to quantify the density of a solution, where a lower value suggests that a solution is more "crowded" by other solutions. This approach does not require the specification of any user-defined parameters for maintaining diversity. A pseudo-code description of the crowding distance procedure is presented in Algorithm 7.4, where $\boldsymbol{P}(i)|o$ represents the i^{th} individual in the population ranked with respect to objective o, while f_o^{max} and f_o^{min} are respectively the maximum and minimum values achieved by individuals along the axis of the o^{th} objective function. The worst-case computational complexity of this procedure is $\mathcal{O}(MN \log N)$ [41]. In contrast, the complexity of the diversity conservation mechanism used in the NSGA-I is $\mathcal{O}(MN^2)$ [41, 134].

In order to guide the selection process of the algorithm towards a uniformly spread approximately

Algorithm 7.3: Fast non-dominated sorting algorithm (FNSA) [41]

Input : A population $\boldsymbol{P}$ containing N solution vectors $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$ in decision space together with the corresponding vectors in objective space.

Output: The set of non-dominated fronts $\mathcal{F}_1, \ldots, \mathcal{F}_F$ partitioning the solutions in $\boldsymbol{P}$.

$\mathcal{F}_1 \leftarrow \emptyset$
forall $\boldsymbol{p} \in \boldsymbol{P}$ **do**
  $\mathcal{D}_{\boldsymbol{p}} \leftarrow \emptyset$
  $n^c_{\boldsymbol{p}} = 0$
  forall $\boldsymbol{q} \in \boldsymbol{P}$ **do**
    if $\boldsymbol{p} \succ \boldsymbol{q}$ **then**
      $\mathcal{D}_{\boldsymbol{p}} \leftarrow \mathcal{D}_{\boldsymbol{p}} \cup \{\boldsymbol{q}\}$
    else if $\boldsymbol{q} \succ \boldsymbol{p}$ **then**
      $n^c_{\boldsymbol{p}} \leftarrow n^c_{\boldsymbol{p}} + 1$
  if $n^c_{\boldsymbol{p}} = 0$ **then**
    $\boldsymbol{p}^{rank} \leftarrow 1$
    $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{\boldsymbol{p}\}$
$\ell \leftarrow 1$
while $\mathcal{F}_\ell \neq \emptyset$ **do**
  $\mathcal{G} \leftarrow \emptyset$
  forall $\boldsymbol{p} \in \mathcal{F}_\ell$ **do**
    forall $\boldsymbol{q} \in \mathcal{D}_{\boldsymbol{p}}$ **do**
      $n^c_{\boldsymbol{q}} \leftarrow n^c_{\boldsymbol{q}} - 1$
      if $n^c_{\boldsymbol{q}} = 0$ **then**
        $\boldsymbol{q}^{rank} \leftarrow \ell + 1$
        $\mathcal{G} \leftarrow \mathcal{G} \cup \{\boldsymbol{q}\}$
  $\ell \leftarrow \ell + 1$
  $\mathcal{F}_\ell \leftarrow \mathcal{G}$

Algorithm 7.4: Crowding distance assignment algorithm [41]

Input : A population $\boldsymbol{P}$ of size N containing solution vectors $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N\}$ and the corresponding performance vectors in the M-dimensional objective space.

Output: The crowding distance for each solution in $\boldsymbol{P}$.

forall $i \in N$ **do**
  $\boldsymbol{p}^{dist}_i \leftarrow 0$
forall $o = 1, \ldots, M$ **do**
  $\boldsymbol{P} \leftarrow sort(\boldsymbol{P}, o)$
  $\boldsymbol{p}^{dist}_{\boldsymbol{P}(1)|o} \leftarrow \infty$
  $\boldsymbol{p}^{dist}_{\boldsymbol{P}(N)|o} \leftarrow \infty$
  forall $i = 2, \ldots, N-1$ **do**
    $\boldsymbol{p}^{dist}_i \leftarrow \boldsymbol{p}^{dist}_i + (\boldsymbol{P}(i+1)|o - \boldsymbol{P}(i-1)|o)/(f^{max}_o - f^{min}_o)$

Pareto optimal front, a so-called *crowded-comparison operator* $\prec_C$ is employed. This operator assesses the rank as well as the crowding distance, as described above, in order to determine superiority between any two solutions, which is used as the selection operator of the algorithm. The rank serves as the first assessment criterion, where a solution achieving a lower rank value is considered superior. If two solutions achieve the same rank value, then the solution achieving the higher crowding distance is considered superior. Mathematically,

$$\boldsymbol{p} \succ_C \boldsymbol{q} \quad \begin{cases} \text{if} & \boldsymbol{p}^{rank} < \boldsymbol{q}^{rank} \\ \text{or} & \boldsymbol{p}^{rank} = \boldsymbol{q}^{rank} \text{ and } \boldsymbol{p}^{dist} > \boldsymbol{q}^{dist}, \end{cases}$$

and the computational complexity of sorting on the operator $\prec_C$ is $\mathcal{O}(N \log N)$ [41]. In this way, the search will favour the exploration of solutions in less crowded regions of the solution space which, in turn, leads to a more uniformly spaced non-dominated front.

Overall, the NSGA-II works as follows. The algorithm is initialised by randomly creating an initial parent population $\boldsymbol{P}_0$ of size N and is iterated until a pre-specified number of generations is achieved. Generation t begins with an initial parent population $\boldsymbol{P}_t$ of size N. The FNSA is called, and each solution is assigned a fitness level proportional to its rank. The usual genetic operators, such as tournament selection, crossover and mutation are then applied to create an offspring population $\boldsymbol{Q}_t$ of size N, after which the parent and child individuals are combined to form a larger intermediate population $\boldsymbol{R}_t = \boldsymbol{P}_t \cup \boldsymbol{Q}_t$ of size $2N$. This intermediate population is then sorted and ranked into non-dominated fronts again, after which the next population of candidate solutions $\boldsymbol{P}_{t+1}$ is generated by first including the solutions in the first non-dominated front $\mathcal{F}_1$, then the solutions in the second non-dominated front $\mathcal{F}_2$, and so on, until a population of size N is reached. Because all individuals in the last included front most likely cannot all be selected to add up to the pre-required population size N, they are ranked in descending order with respect to their crowding distances and the best solutions are chosen one by one until the desired size N is reached for the new population. An illustration of the operation of the NSGA II is shown in Figure 7.16, a flowchart of the algorithm is presented in Figure 7.17 and a pseudo-code description of the algorithm is given in Algorithm 7.5.

7.6.2 Generating an initial population

There is general consensus in the literature that an initial population which is too small has a higher chance of guiding the search towards poor solutions, while an initial population which is too large may incur an unrealistically large computational burden in order to find a good set of non-dominated solutions [24, 79, 169]. According to Diaz-Gomez *et al.* [45], finding a good initial population combined with the optimal population size is a difficult problem, and there exists no general rule for this purpose which can be applied to every type of problem. They acknowledge that the search space, fitness function, diversity, problem difficulty and selection pressure are all factors that may and should also be considered in order to create a good initial population.

The notion of *subspecies* is introduced to fulfil the role of a *state*, as described in §7.5.3. A subspecies is a subpopulation containing all individuals in a specific state. The process of creating an initial population for the MLE response selection problem primarily evolves around the distribution of subspecies, which was found, after extensive numerical experimentation, to have a significant impact on the nature and distribution of subspecies in future generations. A subspecies is said to become *extinct* if the number of individuals in this subspecies is reduced to zero from one generation to the next one. Extinctions are better avoided, particularly during

Algorithm 7.5: The Non-dominated Sorting Genetic Algorithm II (NSGA-II) [41, 90]

Input : A multiobjective optimisation problem with sets of variables, constraints and objective functions, the size of the population N and the maximum number of generations G_{max}.

Output: A non-dominated set of solutions $\tilde{\mathcal{P}}$ in decision space and the corresponding set of performance vectors $\tilde{\mathcal{F}}$ in objective space.

Generate a random feasible initial population $\boldsymbol{P}_0$ of size N
Sort the individuals in $\boldsymbol{P}_0$ into their respective non-dominated fronts using Algorithm 7.3 and assign them a rank
Assess the crowding distance of each solution in $\boldsymbol{P}_0$ using Algorithm 7.4
Generate an initial offspring population $\boldsymbol{Q}_0$ of size N using tournament selection; favour individuals competing for reproduction first based on their ranks and then on their crowding distances
$t \leftarrow 0$
while $t < G_{max}$ **do**
    $\boldsymbol{R}_t \leftarrow \boldsymbol{P}_t \cup \boldsymbol{Q}_t$
    Sort $\boldsymbol{R}_t$ into non-dominated fronts $\mathcal{F}_1, \ldots, \mathcal{F}_F$ using Algorithm 7.3
    $\boldsymbol{P}_{t+1} \leftarrow \emptyset$
    $\ell \leftarrow 1$
    while $|\boldsymbol{P}_{t+1}| < N$ **do**
        if $|\boldsymbol{P}_{t+1}| + |\mathcal{F}_\ell| \leq N$ **then**
            $\boldsymbol{P}_{t+1} \leftarrow \boldsymbol{P}_{t+1} \cup \mathcal{F}_\ell$
        else if $|\boldsymbol{P}_{t+1}| + |\mathcal{F}_\ell| > N$ **then**
            Assess the crowding distance of each solution in $\mathcal{F}_\ell$ similarly to Algorithm 7.4
            $\mathcal{F}_\ell = sort(\mathcal{F}_\ell, \succ_C)$
            $c \leftarrow 1$
            while $|\boldsymbol{P}_{t+1}| + c \leq N$ **do**
                $\boldsymbol{P}_{t+1} \leftarrow \boldsymbol{P}_{t+1} \cup \mathcal{F}_\ell(c)$
                $c \leftarrow c + 1$
        $\ell \leftarrow \ell + 1$
    Assess the crowding distance of each solution in $\boldsymbol{P}_{t+1}$
    Generate an offspring population $\boldsymbol{Q}_{t+1}$
    $t \leftarrow t + 1$
$\mathcal{P} \approx \tilde{\mathcal{P}} = \mathcal{F}_1$

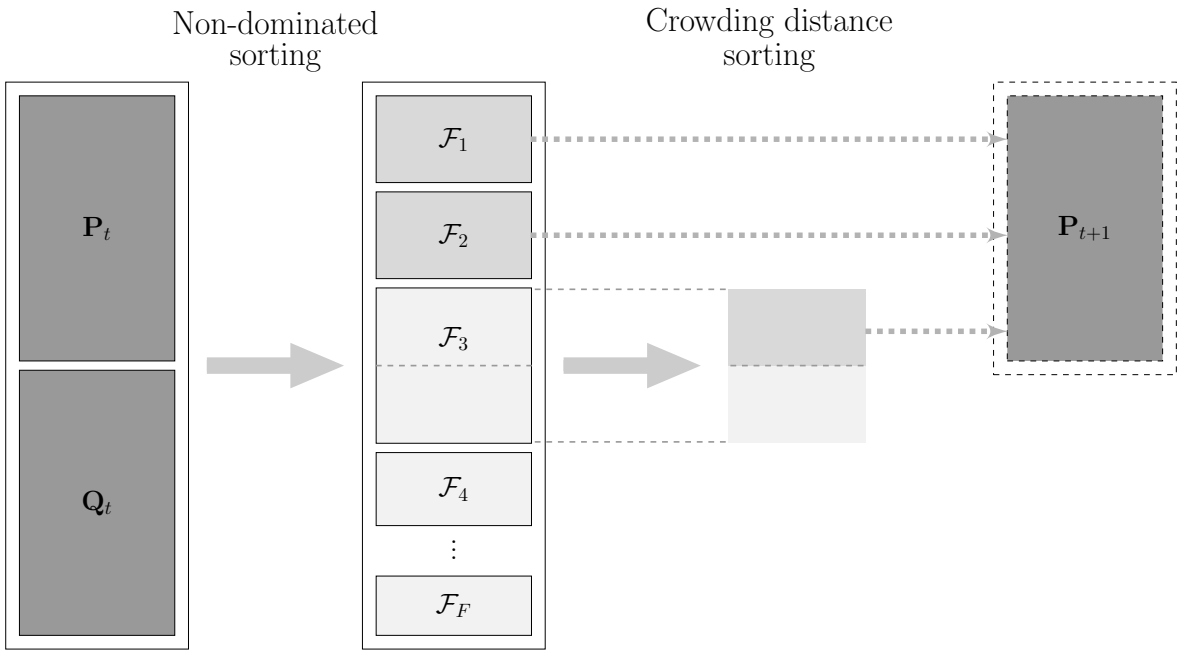


FIGURE 7.16: *Sequence of operations carried out in the NSGA-II.*

early stages of the evolution process, as they may significantly affect the diversity status of the final population and promote premature convergence.

The proposed crossover operator (presented later in this section) allows for inter-subspecies reproduction, but does not allow for offspring solutions from different subspecies to that of their parents to be generated. Indeed, not only does this idea go strongly against nature, but may also create offspring solutions that differ too much from their parents, hence defeating the purpose of passing on strong genetic material from the parent population through inheritance. Three incentives are employed in an attempt to reduce the extent of the problem of extinction: (1) a large initial population is generated with subspecies that are assigned large enough sizes; (2) an immigration operator is introduced in which the immigrants are individuals which belong to subspecies that have become extinct or are dangerously close to becoming so; and (3) the mutation operator is configured to apply inter-state transformations.

In respect of solution transformations using genetic operators, individuals are represented by means of *chromosome strings* composed of cells, as described in §7.3. If the size of the initial population is deemed too small or if the subspecies vary too much in size from one another, then the mutation operator in the proposed genetic algorithm methodology is configured to use inter-state transformations (see §7.5.3) to mutate solutions according to a certain probability in order to promote diversity throughout the evolution process, thereby addressing the problem of extinction described above. Else, a simple intra-route transformation is configured for the purpose of mutation. Furthermore, it is customary to use tournament selection for developing the mating pool when applying the NSGA-II [41, 126], where a certain number of individuals are selected stochastically from the current population, and the fittest ones (assessed using the crowded comparison operator, as described in the previous section) are paired for reproduction.

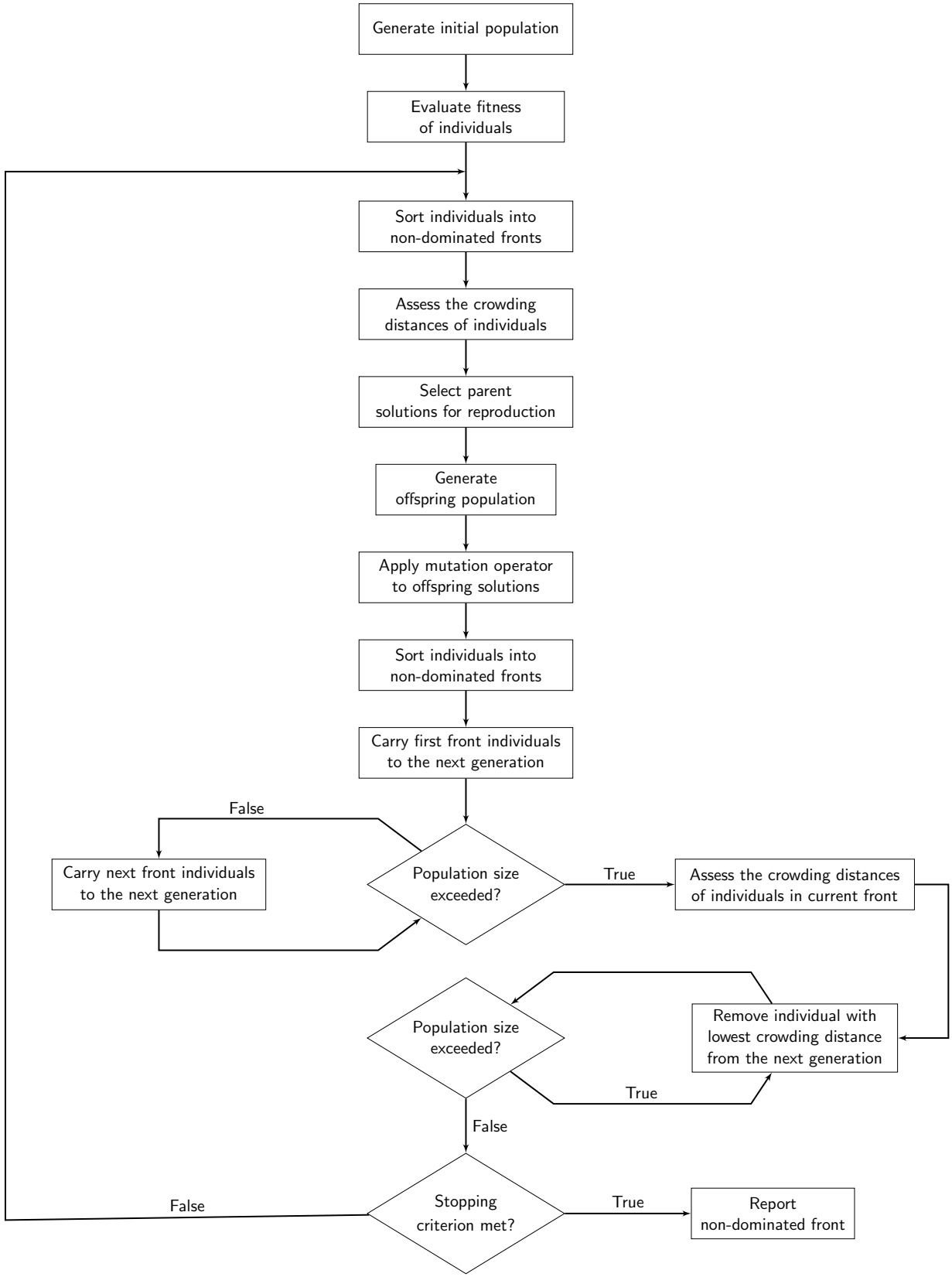


FIGURE 7.17: *Flow of events in the NSGA-II.*

7.6.3 Crossover operator configuration

The main property rendering genetic solution representations convenient is that their parts are easily aligned due to their fixed size which, in turn, facilitates simple crossover operations [81]. Variable solution length representations may also be used, but crossover implementation is more complex in this case. Moreover, according to Nazif and Lee [106], it may often be necessary with genetic algorithms not to generate the entire offspring population by means of crossover operations. In their paper, for instance, they implement a crossover probability of 0.75 for every pair of parents and a swap operator (*i.e.* one-to-one exchange) otherwise. In [119], eight different crossover operators obtained from various previous publications are analysed in respect of transforming solution strings for VRPs, none of which deal with variable chromosome string lengths that are applicable to VRPs with customer profits, and this is the same in other, related publications. Consequently, due to the lack of adequate crossover operators in the literature, a functional crossover procedure for successfully creating feasible offspring solutions is proposed in this section for solving the MLE response selection problem by means of the NSGA-II.

Given two selected parent solutions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, let $\mathcal{R}_{\boldsymbol{p}_1} = \{r_1(\boldsymbol{p}_1), \ldots, r_{|\mathcal{V}^r|}(\boldsymbol{p}_1)\}$, where $r_k(\boldsymbol{p}_1)$ represents the number of VOIs in route $k \in \mathcal{V}^r$ of the reduced string of parent solution $\boldsymbol{p}_1$. Similarly, let $\mathcal{R}_{\boldsymbol{p}_2} = \{r_1(\boldsymbol{p}_2), \ldots, r_{|\mathcal{V}^r|}(\boldsymbol{p}_2)\}$. The proposed crossover procedure consists of two sub-processes, where the second sub-process is only initiated if the first crossover sub-process fails (with respect to feasibility) after a certain number of mating attempts.

The first crossover sub-process may be described as an adapted version of the traditional cut-and-splice crossover operator (as described in §2.4.4), which attempts to swap genetic material from both parents as follows. First, the delimiter cells are removed from their reduced solution strings to create two chromosome strings containing VOI cells only. A *marker function* $f_M : (|\boldsymbol{p}_1|, |\boldsymbol{p}_2|) \rightarrow [1, \min\{|\boldsymbol{p}_1|, |\boldsymbol{p}_2|\}]$ is specified by the user, where $|\boldsymbol{p}_1|$ refers to the number of cells in the chromosome string of $\boldsymbol{p}_1$ and $|\boldsymbol{p}_2|$ to the number of cells in the chromosome string of $\boldsymbol{p}_2$. This user-defined step-function generates an integer, called the *marker* and denoted by M, in the specified range. The nature of the marker depends on the size of the parent strings, $|\boldsymbol{p}_1|$ and $|\boldsymbol{p}_2|$, as well as on their difference in size. Two offspring solutions $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$ are constructed by swapping partial strings from the parents (similarly to a cut-and-splice crossover operator), where the cut is defined by the marker M. A test is then carried out to ascertain whether any of the two offspring solutions contain duplicate cells in their chromosomes. If one or both of them do, then the marker value is reduced by one, and the cut-and-splice crossover process is repeated for $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, using the updated marker value. This crossover sub-process is terminated when two offspring chromosomes are formed which do not contain duplicate entries. Converting the offspring chromosome strings back into a solution format is achieved by configuring $\mathcal{R}_{\boldsymbol{o}_1} = \mathcal{R}_{\boldsymbol{p}_2}$ and $\mathcal{R}_{\boldsymbol{o}_2} = \mathcal{R}_{\boldsymbol{p}_1}$. These offspring solutions are then tested further for overall feasibility. If the marker value reaches zero, however, or if one or two of the offspring solutions is classified as infeasible with respect to overall feasibility (which includes the reinsertion and end-of-route assignment sub-transformations performed on these reduced offspring solution strings), then the first sub-process of this crossover procedure is aborted, and the second sub-process is initiated.

An illustrative description of a successful procedure conducted during the first sub-process of this crossover operator is presented in Figure 7.18. Here, $|\boldsymbol{p}_1| = 7$ and $\mathcal{R}_{\boldsymbol{p}_1} = \{2, 3, 2\}$, while $|\boldsymbol{p}_2| = 5$ and $\mathcal{R}_{\boldsymbol{p}_2} = \{1, 2, 2\}$. For demonstration purposes, assume that the original marker value was specified as $M = 3$. Then, performing a cut-and-splice crossover procedure on the two parent chromosomes using this marker clearly results in an infeasible offspring chromosome (with respect to testing for duplicate entries), as the cell containing entry 5 is duplicated in the chromosome of the second offspring. However, as depicted in this example, lowering the marker

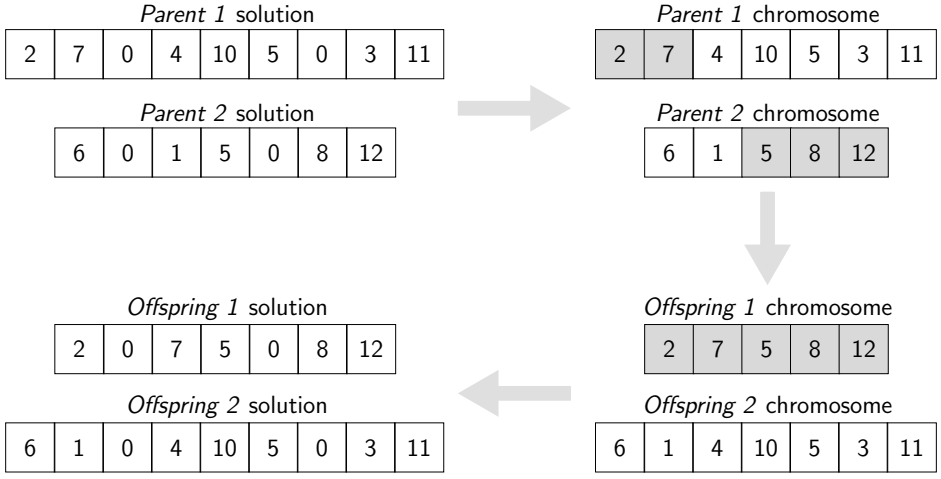


FIGURE 7.18: *Procedure (successfully) completed during the first sub-process of the proposed crossover operator.*

value by one (*i.e.* setting $M = 2$) produces two feasible offspring chromosomes (again, only with respect to testing for duplicate entries), which may then be converted back into an appropriate solution format by reinserting the delimiter cells with the sets $\mathcal{R}_{\boldsymbol{O}_1}$ and $\mathcal{R}_{\boldsymbol{O}_2}$.

The second (backup) sub-process of the crossover procedure was inspired by the natural concept of *asexual reproduction*, in which an offspring solution is created from a single organism by inheriting (presumably good) genes from a single parent solution. This sub-process is therefore conducted for each parent solution independently. In this procedure, *clones* are defined as pairs of individuals which possess identical chromosome strings. Without loss of generality, this second sub-process is described for offspring $\boldsymbol{o}_1$ only, and works as follows. First, let $\boldsymbol{o}_1$ be configured as a clone of $\boldsymbol{p}_2$, and let $\mathcal{R}_{\boldsymbol{O}_1} = \mathcal{R}_{\boldsymbol{p}_2}$. Two elements are randomly selected from the set $\mathcal{R}_{\boldsymbol{O}_1}$ such that at least one of these elements is larger than, or equal to, one. One of these two elements is then randomly selected (unless one of the elements is zero, in which case the other one is selected) and one is subtracted from it, while one is added to the other element. These new numbers of routes are implemented in order to convert the offspring chromosome to an offspring solution, and a test is carried out for overall feasibility. If the offspring solution is infeasible, $\mathcal{R}_{\boldsymbol{O}_1}$ is reset to $\mathcal{R}_{\boldsymbol{p}_2}$ and the process is repeated by selecting two random elements from the set $\mathcal{R}_{\boldsymbol{O}_1}$, and so forth. In genetic algorithms, the process by which attempts are made to convert infeasible offspring solutions into feasible ones by means of a local search operator falls in the class of *education* operators (see §2.4.1). This sub-process is terminated once a complete feasible offspring solution has been generated (*i.e.* once it has successfully been educated), or a pre-determined number of attempts O_{max} have been made. In the former case, the offspring solution is added to the offspring population, while in the latter case the overall crossover procedure is aborted.

An illustrative description of a procedure conducted during the second sub-process of the crossover operator is presented in Figure 7.19. Here, the offspring chromosome string is initially identical to the parent chromosome string, and $\mathcal{R}_{\boldsymbol{O}} = \mathcal{R}_{\boldsymbol{P}} = \{1, 2, 2\}$. The first and third elements of this set are then randomly selected and one is subtracted from the third element, while one is added to the first element, resulting in $\mathcal{R}_{\boldsymbol{O}} = \{2, 2, 1\}$. A pseudo-code description of the overall crossover procedure is provided in Algorithm 7.6.

Algorithm 7.6: Crossover procedure outline

Input : Two parent solutions $\boldsymbol{p}_1, \boldsymbol{p}_2$ with respective route sizes $\mathcal{R}_{\boldsymbol{p}_1}$ and $\mathcal{R}_{\boldsymbol{p}_2}$, a marker function f_M, and a maximum number of offspring generation attempts O_{max}.

Output: Possibly feasible offspring solutions $\boldsymbol{o}_1$ and $\boldsymbol{o}_2$.

1 $M \leftarrow f_M(|\boldsymbol{p}_1|, |\boldsymbol{p}_2|)$
2 $\boldsymbol{p}_1 \leftarrow [\boldsymbol{p}_{11}\ \boldsymbol{p}_{12}]$ with $[\boldsymbol{p}_{11}] = [\boldsymbol{p}_1(1), \ldots, \boldsymbol{p}_1(M)]$ and $[\boldsymbol{p}_{12}] = [\boldsymbol{p}_1(M+1), \ldots, \boldsymbol{p}_1(|\boldsymbol{p}_1|)]$
3 $\boldsymbol{p}_2 \leftarrow [\boldsymbol{p}_{21}\ \boldsymbol{p}_{22}]$ with $[\boldsymbol{p}_{21}] = [\boldsymbol{p}_2(1), \ldots, \boldsymbol{p}_2(M)]$ and $[\boldsymbol{p}_{22}] = [\boldsymbol{p}_2(M+1), \ldots, \boldsymbol{p}_2(|\boldsymbol{p}_2|)]$
4 $\boldsymbol{o}_1 \leftarrow [\boldsymbol{p}_{11}\ \boldsymbol{p}_{22}]$
5 $\boldsymbol{o}_2 \leftarrow [\boldsymbol{p}_{21}\ \boldsymbol{p}_{12}]$
6 **if** *either* $\boldsymbol{o}_1$ *or* $\boldsymbol{o}_2$ *contain duplicates* **then**
7 $M \leftarrow M - 1$
8 **if** $M \geq 0$ **then**
9 Go to step 2
10 **else**
11 $\boldsymbol{o} \leftarrow \boldsymbol{o}_1$
12 $\boldsymbol{p} \leftarrow \boldsymbol{p}_2$
13 $t \leftarrow 1$
14 **while** *(*$t \leq O_{max}$*) and (*$\boldsymbol{o}$ *is infeasible)* **do**
15 $\mathcal{R}_{\boldsymbol{o}} \leftarrow \mathcal{R}_{\boldsymbol{p}}$
16 Generate random numbers r_1, r_2 such that either $\mathcal{R}_{\boldsymbol{o}}(r_1) \geq 1$ or $\mathcal{R}_{\boldsymbol{o}}(r_2) \geq 1$
17 **if** $r_1 \geq 1$ **then**
18 $\mathcal{R}_{\boldsymbol{o}}(r_1) \leftarrow \mathcal{R}_{\boldsymbol{o}}(r_1) - 1$
19 $\mathcal{R}_{\boldsymbol{o}}(r_2) \leftarrow \mathcal{R}_{\boldsymbol{o}}(r_2) + 1$
20 $t \leftarrow t + 1$
21 **else**
22 $\mathcal{R}_{\boldsymbol{o}}(r_1) \leftarrow \mathcal{R}_{\boldsymbol{o}}(r_1) + 1$
23 $\mathcal{R}_{\boldsymbol{o}}(r_2) \leftarrow \mathcal{R}_{\boldsymbol{o}}(r_2) - 1$
24 $t \leftarrow t + 1$
25 **if** $\boldsymbol{o}$ *is feasible* **then**
26 Append $\boldsymbol{o}$ to the offspring population
27 **if** *The education operator has been carried onto both offspring solutions* **then**
28 STOP
29 **else**
30 $\boldsymbol{o} \leftarrow \boldsymbol{o}_2$
31 $\boldsymbol{p} \leftarrow \boldsymbol{p}_1$
32 Go to step 13
33 **else**
34 Go to step 11

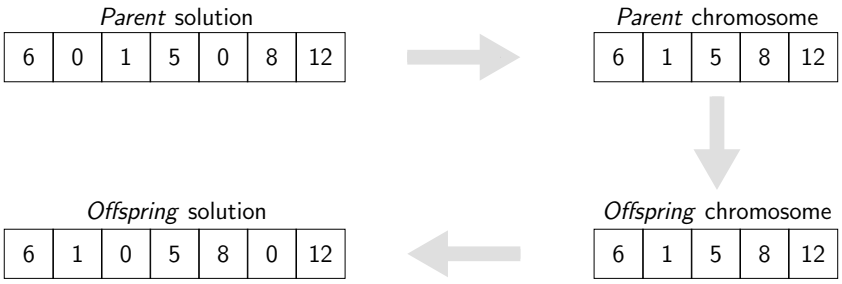


FIGURE 7.19: *Procedure (successfully) completed during the second sub-process of the proposed crossover operator.*

7.7 Design of a hybrid metaheuristic

Due to the high complexity and dynamic nature of the MLE response selection problem, standard metaheuristics may not perform well enough in terms of generating a high-quality set of non-dominated solutions under pressure (*i.e.* within a limited computational time). One way of alleviating this problem is to design a hybrid algorithm which combines two or more metaheuristics in an attempt to better guide the search process. An idea is proposed in this section for the design of a hybrid metaheuristic combining the population-based, non-dominated sorting procedure and diversity preservation rule of the NSGA-II with the Metropolis rule of acceptance and the notion of archiving of the multiobjective simulated annealing method.

As discussed in §2.4.4, a mutation operator typically modifies an offspring solution stochastically in such a way that the result of the transformation is close to its original chromosome form. The aim of mutation is to maintain an acceptable level of diversity among population individuals at every generation. If, however, the mutation process is not random, especially during the early stages of the search, and only improving moves are accepted, the diversity status of the population may become compromised, causing the search to run the risk of converging prematurely. Premature convergence causes large regions of the search space to remain unexplored, which ultimately results in a non-uniformly spread non-dominated front.

Moreover, as mentioned, the proportion of mutated individuals in the offspring population is controlled using a mutation rate parameter. A mutation operator functioning according to a reasonably high mutation rate is expected to produce the desired results in terms of the preservation of diversity. Setting the mutation rate too high, however, may result in individuals being produced almost independently from the genetic material passed on to them from their parents, which would cause the algorithmic progress basically to turn into a primitive random walk of the search space.

The origin of the proposed hybrid approach evolves around the idea that mutation moves which are accepted stochastically at first, but increasingly only upon improvement later on during the search, may increase the average fitness of individuals at every generation with very little added complexity, while significantly reducing the risk of premature convergence. A cooling schedule $T_1, \ldots, T_C$ with corresponding epoch durations $L_1, \ldots, L_C$ is constructed in a similar way to that employed in Algorithm 7.2. The outer loop consists of the NSGA-II as well as a cooling schedule, while an inner loop conducts a procedure based on the Metropolis algorithm. Each time an offspring solution is selected for mutation, the selected mutation operator attempts to improve the solution a finite number of times, or exit the loop if it fails to do so. The archive of the simulated annealing method is configured as the first non-dominated front of the current

generation, which is used (as reference in respect of solution quality) to assess the change in energy between any two mutated solutions as part of the Metropolis acceptance rule. A pseudo-code description of the proposed hybrid metaheuristic is provided in Algorithm 7.7.

Algorithm 7.7: Hybrid metaheuristic (NSGAII–AMOSA) outline

Input : A multiobjective optimisation problem with sets of variables, constraints and objective functions, a cooling schedule $\{T_1, \ldots, T_C\}$ with epoch durations $\{L_1, \ldots, L_C\}$, a mutation rate p_m, the size of the population N, the maximum number of iterative attempts in the simulated annealing sub-loop I_{max} and the maximum number of generations G_{max}.

Output: A non-dominated set of solutions $\tilde{\mathcal{P}}$ in decision space and the corresponding set of performance vectors $\tilde{\mathcal{F}}$ in objective space.

Perform steps 1–4 of Algorithm 7.5
$t \leftarrow 0$
Initialise cooling schedule epoch $c \leftarrow 1$
while $t < G_{max}$ **do**
  while $t \leq \sum_{a=1}^{c} L_{a-1} + L_c$ **do**
    Perform steps 7–25 of Algorithm 7.5
    Generate an offspring population $\boldsymbol{Q}_{t+1}$
    Set archive $\mathcal{A} \leftarrow \mathcal{F}_1$
    forall $\boldsymbol{o} \in \boldsymbol{Q}_{t+1}$ **do**
      Generate random number $r_1 \in (0, 1)$
      if $r_1 < p_m$ **then**
        $i \leftarrow 1$
        while *($i \leq I_{max}$) and (mutation move has not been accepted)* **do**
          Generate offspring solution $\boldsymbol{o}'$ by mutating current offspring solution $\boldsymbol{o}$
          while $\boldsymbol{o}'$ *is infeasible* **do**
            Mutate offspring solution $\boldsymbol{o}$
          Generate random number $r_2 \in (0, 1)$
          if $r_2 < \min\left\{1, \exp\left(\frac{-\Delta_E(\boldsymbol{o}', \boldsymbol{o})}{T_c}\right)\right\}$ **then**
            Accept move
            $\boldsymbol{Q}_{t+1} \leftarrow \boldsymbol{Q}_{t+1} \cup \{\boldsymbol{o}'\}$
            $\boldsymbol{Q}_{t+1} \leftarrow \boldsymbol{Q}_{t+1} \backslash \{\boldsymbol{o}\}$
          else
            $i \leftarrow i + 1$
    $t \leftarrow t + 1$
  $c \leftarrow c + 1$
$\mathcal{P} \approx \tilde{\mathcal{P}} = \mathcal{F}_1$

7.8 Simulated annealing methodology for parallel optimisation

Due to the potentially very large domain size of an MLE response selection scenario, it is believed that, realistically, even the most suited sequential solution search configurations will not be capable to meet the requirements of a "good" non-dominated set of solutions (see §7.4) when subjected to a strict computational budget.

To remedy to this problem, it is advocated that any real-life MLE response selection DSS should make use of parallel computing[5]. Indeed, as a result of the increase in the availability of memory and CPU power, parallel optimisation solution search techniques are expected to guide the search closer toward the true Pareto front in terms of proximity, diversity, and extremal values. In particular, given a fixed period of computational time, larger subsets of the feasible decision space may then be uncovered when parallel computing is employed. Algorithms applied within a parallel computing paradigm are therefore expected to perform much better than their sequential computing counterparts. The manner in which the algorithmic tasks are distributed amongst the various processing units, as well as the manner in which these units interact with one another, is however expected to have a significant impact on the performance results.

In order to simulate the benefits of using parallel computing as a means to solve a given MLE response selection problem instance, two simple multistart simulated annealing algorithms are put forward in this section, namely an *interactive multistart archived simulated annealing* and an *augmented multistart archived simulated annealing*. The configurations of these algorithms were both inspired by the (unistart) archived simulated annealing algorithm described in §7.5.

In the first algorithm above, each of N *ancillary processors* is assigned a specific annealing schedule and solution transformation parameters. An additional processor is also responsible for storing and keeping track of a centralised archive $\mathcal{A}$ in real time. Each of the ancillary processors is first populated with a single feasible initial solution. These solutions are then merged in the central archive, and a test for Pareto dominance is conducted in order to discard possibly dominated solutions from this archive. Following this, the respective cooling schedules are simultaneously initiated in each ancillary processor, after which each of these processors independently conducts its own multiobjective archived simulated annealing process, but collectively using the single, central archive as a means to develop the non-dominated set of solutions iteratively. More specifically, at every iteration, each ancillary processor takes a snapshot of the current archive from the central processor in order to evaluate the acceptance probability function of its generated neighbouring solution, as well to test (if the move has been accepted) whether this neighbouring solution is non-dominated with respect to the current state of the archive (appending it to this set if that is the case). This process is terminated when a pre-determined maximum number of iterations has been reached. Because certain parametric configurations may generally find feasible neighbouring solutions quicker than others, it is assumed, for the sake of simplicity, that all ancillary processors are iterated according to a central clock rather than in their own time (although it is acknowledged that individual clock times would work just as well). In other words, the next iteration may only be triggered once every ancillary processor has successfully generated and assessed a feasible neighbour solution. As this sub-process is not carried out sequentially amongst the ancillary processors, but rather simultaneously, however, it may be the case that the final archive contain a certain number of dominated solutions[6]. A filtering process is therefore conducted on this final archive in order to discard these undocumented dominated solutions. A pseudo-code description of this algorithmic solution approach is given in Algorithm 7.8.

Although significantly more straightforward than its multistart counterpart, the second algorithm put forward is, nevertheless, expected to be just as effective, if not more. The main and only difference in this parallel search alternative is that, instead of relying on a central processor to store and update a global archive in synchronisation with the ancillary processors, each of

[5]The (fixed) costs of investing in powerful parallel processors is expected to be negligible compared to the overall budget allocated to the MLE operations of a coastal nation.

[6]Suppose, for example, that one of the ancillary processors appends a non-dominated solution $\boldsymbol{x}$ to the central archive, while a different processor also appends a seemingly non-dominated solution $\boldsymbol{y}$ to the archive, both during the same iteration. Then, consider the case where $\boldsymbol{x} \succ \boldsymbol{y}$.

Algorithm 7.8: Interactive multistart archived simulated annealing outline

Input : A multiobjective problem with sets of variables, constraints and objective functions, N independent parallel processors with respective annealing schedule and solution transformation procedures stored in the parametric configuration sets $(\Omega_1, \Omega_2 \ldots, \Omega_N)$, a maximum number of iterations I_{max}, and an $N+1^{th}$ processor storing a central archive.

Output: A non-dominated set of solutions $\tilde{\mathcal{P}}$ in decision space and the corresponding set of performance vectors $\tilde{\mathcal{F}}$ in objective space.

```
1  Generate a set of initial feasible solutions X = {x_1, ..., x_N}
2  Initialise archive A = X and remove any dominated solution from A
3  Initialise cooling schedule in each ancillary processor
4  Initialise iterations counter t ← 1
5  while t ≤ I_max do
6      forall p ∈ N do
7          Reform steps 7–23 of Algorithm 7.2 using parameter set Ω_p
8          Lower temperature according to the cooling schedule in Ω_p
9      t ← t + 1
10 P ≈ P̃ = A
```

the N processors rather stores and updates its own archive and operates in its own real time. This approach is therefore equivalent to conducting N simultaneous, yet independent (*i.e.* non-interacting) unistart multiobjective archived simulated annealing search algorithms. Once all processors have terminated their respective search processes, however, their archives are combined into a global, augmented archive, following which duplicate and dominated solutions are discarded from this superset to form a valid set of non-dominated solutions with corresponding solution mappings in objective space. The use of an additional processor is not required for the sole process of merging the individual archives at the end of the algorithm; this low computational complexity task may instead simply be conducted on any one of the (now idle) processors. A pseudo-code description of this algorithmic solution approach is given in Algorithm 7.9.

7.9 Chapter summary

In the first section of this chapter, a simplified algebraic description was derived for approximating interception points between MLE resources and VOIs. A method for assigning end-of-route assignments to MLE resources at the end of their missions was then proposed in §7.2, and this was followed in §7.3 by a generic approach towards encoding solution strings in the context of an MLE response selection problem as well as performing overall solution transformation procedures, and a discussion on general Pareto front approximation in §7.4. A dominance-based archived multiobjective simulated annealing approach adapted to the MLE response selection problem was described in depth in §7.5 while, in the succeeding section, the NSGA-II was described as an alternative solution process for solving this problem. In addition, an idea was proposed in §7.7 for a hybrid metaheuristic combining the population-based, non-dominated sorting procedure and diversity preservation rule of the NSGA-II with the Metropolis rule of acceptance and archiving of the multiobjective simulated annealing method. Finally, two multistart archived simulated annealing methodologies adapted to a parallel computation paradigm were put forward in §7.8.

Algorithm 7.9: Augmented multistart archived simulated annealing outline

Input : A multiobjective problem with sets of variables, constraints and objective functions, a population of N individual solutions on N parallel processors with respective annealing schedule and solution transfomation procedures stored in the parametric configuration sets $(\Omega_1, \Omega_2 \ldots, \Omega_N)$, and a maximum number of iterations I_{max}.

Output: A non-dominated set of solutions $\tilde{\mathcal{P}}$ in decision space and the corresponding set of performance vectors $\tilde{\mathcal{F}}$ in objective space.

1 **forall** $\boldsymbol{i} \in 1, \ldots, N$ **do**
2 Generate initial feasible solution $\{\boldsymbol{x}\}$
3 Initialise processor archive $\mathcal{A}_i = \{\boldsymbol{x}\}$
4 Reform steps 3–27 of Algorithm 7.2 using parameter set Ω_i
5 $\mathcal{A} = \cap_{i \in N} \mathcal{A}_i$
6 **forall** $\boldsymbol{y} \in \mathcal{A}$ **do**
7 **forall** $\boldsymbol{x} \in \mathcal{A}$ **do**
8 **if** $\boldsymbol{x} \succ \boldsymbol{y}$ **then**
9 $\mathcal{A} \leftarrow \mathcal{A} \backslash \{\boldsymbol{y}\}$
10 $\mathcal{P} \approx \tilde{\mathcal{P}} = \mathcal{A}$

In the following chapter, two hypothetical scenarios of varying complexities are considered, and the optimisation methodologies introduced in this chapter are tested in these contexts in respect of their capabilities of generating high-quality solutions to instances of the MLE response selection problem.

CHAPTER 8

System Validation

Contents

In order to validate the MLE response selection DSS proposed in Chapter 3 and its incorporation into the MLE system architecture of the same chapter, the aim in this chapter is to gain a better practical understanding of the mathematical modelling and solution search subsystems by testing these in the context of two hypothetical MLE response selection scenarios. More specifically, the modelling of the problem instance embodied in these scenarios is achieved by utilising a subset of the features described in Chapters 4 and 5, while solving the problem instance is achieved by applying the optimisation methodologies proposed in Chapter 7.

This chapter is structured as follows. A method for assessing the performance of stochastic optimisation techniques is first described in 8.1, with particular focus on the well-known hypervolume measure, after which an evaluation involving the domain size of any given MLE response selection problem instance is pursued in §8.2. A lower-complexity hypothetical scenario is then designed in §8.3 within an intermediate decision making MLE response selection paradigm, and this is followed §8.4 by a presentation of numerical results which were obtained by solving the hypothetical problem instance of §8.3 using an array of optimisation techniques. These results are then analysed and interpreted in some detail. In order to test the optimisation methodology component under more strenuous conditions, a subset of these experiments is repeated in the context of a higher-complexity hypothetical scenario in §8.5. Furthermore, recall from Chapter 3 that the proposed DSS also incorporates a post-optimisation solution analysis component, which has not been discussed so far in the dissertation; its purpose is explained in §8.6. The chapter finally closes with a brief summary in §8.7.

8.1 Stochastic multiobjective optimisers performance assessment

The comparison of various optimisation techniques by means of experimentation typically involves the notion of *performance* which, according to Fonseca *et al.* [58], includes both the quality of the incumbent solution or non-dominated front, and the time taken to obtain it. The difficulty in assessing the performance of stochastic optimisation techniques lies in this connection between solution quality and computation time, which may be reflected by a corresponding probability density function. In addition, every statement about the performance of such techniques is also probabilistic in nature.

As discussed in §7.4, the performance of a search technique in the case of multiobjective optimisation problems is typically assessed by considering the distance between the non-dominated front generated and the true Pareto front (which ought to be minimised), the uniformity of the distribution of the non-dominated solutions in objective space (assessment of this criterion is usually based on a certain distance metric) and the extent of the non-dominated front across each objective function (which ought to be maximised). Accurately assessing the respective quality of non-dominated fronts generated by various multiobjective stochastic optimisation techniques within specific time frames, while considering the objective function trade-off values and preferences of the decision maker is not a trivial task, particularly in cases where more than two objective functions are considered [58]. Sound inferences may, however, be made using various forms of quantitative and statistical information obtained from a series of carefully designed experiments in order to carry out such assessments.

Two fundamental approaches for evaluating the performance of multiobjective optimisation techniques are available in the literature, namely the *attainment function* approach and the *indicator* approach [58]. The first approach consists of modelling the outcome of a solution search process, or *run*, as a probability density function in objective space, while the second

summarises the outcome of such a run on the basis of certain quantitative performance measures by conducting a statistical analysis on the distribution of the performance values. Informally speaking, the latter approach is typically used for measuring the total portion of the objective space covered in some sense by the non-dominated front.

If the latter approach is employed, care must be taken when selecting the quality indicator measure, as every indicator possesses a different underlying structure and, consequently, embodies certain assumptions about the decision maker's preferences [58]. It follows that, in any comparative study in which one or more quality indicators are utilised, one may not merely state that a certain optimiser outperforms another one without acknowledging that such a statement may only be claimed under the assumption that the utilised indicator(s) fundamentally reflect the decision maker's preferences.

8.1.1 The hypervolume quality indicator

The *hypervolume indicator*, denoted here by $\overline{H}_{vol}$ and also known as the *S-metric*, is a popular indicator measure used for evaluating the performance of non-dominated fronts. Such an indicator belongs to the class of *unary* quality indicators, which may be defined as mappings from a non-dominated set of solutions to the set of positive real numbers. Its purpose is to calculate the portion of objective space that is collectively dominated by the solutions in the non-dominated front with respect to a specific, pre-defined vector in objective space, known as the *reference point* and chosen in such a way that the non-dominated objective space is bounded by such a point. The hypervolume measure is usually maximised, and attains its maximum if and only if the non-dominated front is equal to the true Pareto front.

Informal description and benefits

Consider the set of non-dominated points $\mathcal{S}_{\boldsymbol{x}} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$ in the solution space of a multiobjective optimisation problem with respective performance vectors $\mathcal{S}_{\boldsymbol{z}} = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^N\}$ in objective space. The hypervolume measure of $\mathcal{S}_{\boldsymbol{x}}$ is then defined as the region dominated by $\mathcal{S}_{\boldsymbol{z}}$ in objective space from a reference point $\bar{\boldsymbol{z}}$ satisfying $\boldsymbol{z}^\ell \succeq \bar{\boldsymbol{z}}$ for all $\ell \in \{1, \ldots, N\}$. An illustration of the hypervolume measure for a bi-objective minimisation problem using the above information is shown in Figure 8.1.

The hypervolume indicator is often favoured as a performance measure because of its ability to capture, in a single scalar measure, both the diversity of non-dominated solutions in objective space and the spread of these solutions across the non-dominated front [159]. Additionally, it possesses very appealing mathematical properties which allow, *inter alia*, for the establishment of important axioms and identities. For instance, any points generated during the search which are not in the non-dominated set of solutions do not contribute to the hypervolume, as the region covered by such a point is completely covered by the region covered by the points dominating it. This property has significant benefits in speeding up the computation of the hypervolume indicator.

Moreover, according to Zitzler *et al.* [168], the hypervolume indicator possesses two further major advantages over other measures. First, it is sensitive to any type of objective score improvements in the sense that, whenever a non-dominated set of solutions dominates another one, then the measure provides a strictly better value for the former set than it does for the latter one. That is, given any two non-dominated sets $\mathcal{D}_1$ and $\mathcal{D}_2$, $\overline{H}_{vol}(\mathcal{D}_1) \geq \overline{H}_{vol}(\mathcal{D}_2)$ whenever $\mathcal{D}_1 \rhd \mathcal{D}_2$ (*i.e.* whenever every element in $\mathcal{D}_2$ is weakly dominated by at least one element in

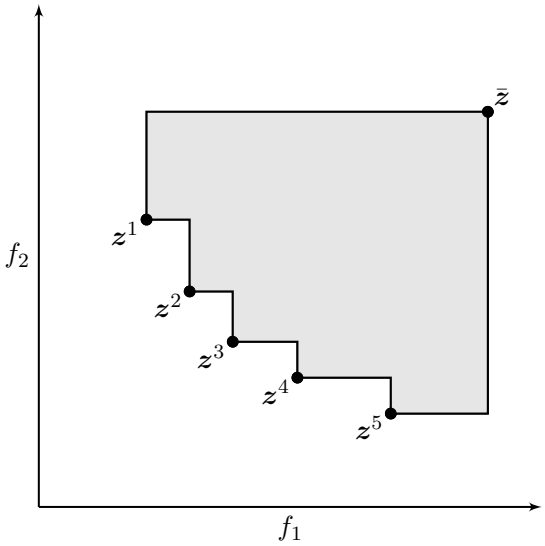


FIGURE 8.1: *Hypervolume of a non-dominated front consisting of the five points* $\mathcal{S}_{\boldsymbol{z}} = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^5\}$, *indicated by the surface area of the shaded region. The objective functions* f_1 *and* f_2 *are both to be minimised and the reference point is* $\bar{\boldsymbol{z}}$.

$\mathcal{D}_1$). Secondly, as a result of this property, this measure guarantees that any non-dominated set achieving the largest possible performance value for a specific problem instance contains all true Pareto-optimal points in objective space.

Finally, the hypervolume indicator has been shown to be capable of detecting differences in extent, proximity to the true Pareto front, and evenness, when comparing two non-dominated fronts. The implications of these properties of the hypervolume indicator therefore allow for the notion of the term *better* to expand to much more than just partial ordering affiliations [58].

Implementation challenges

The hypervolume indicator is, unfortunately, also subject to three main implementation challenges, which must be dealt with carefully [58, 126, 143]. First, its computational complexity is relatively high[1], especially when four or more objectives are implemented. Several approximation approaches have, however, been developed to resolve this drawback, one of which is discussed later in this section.

Secondly, the hypervolume measure calls for non-dominated points in objective space to be normalised in order to ensure that the objectives individually contribute towards the hypervolume calculation in equal proportions. This process requires knowledge of approximate objective bounds across the true Pareto Front. But according to While *et al.* [160], if the true Pareto front maximum/minimum values in each objective are not known, it is good enough to take, for each objective, the best/worst values amongst all non-dominated fronts being compared. Alternatively, such bounds may be closely estimated using methods of reverse engineering. It is, nevertheless, noted that the hypervolume measure is sensitive to the relative scaling of the objectives, particularly with respect to the presence/absence of extremal points across the non-dominated front.

Lastly, this indicator requires the configuration of a reference point (described previously), upon

[1]The exact calculation of the hypervolume is classified as an NP-hard problem [160].

which the metric calculation of the hypervolume is based. It is crucial to select the reference point in such a way so as not to allow certain objectives to contribute more than others towards the hypervolume measure. In practice, various rules-of-thumb exist for choosing the reference point, and many authors recommend the use of the "corner" of an objective space that is a "little bit larger" than the actual objective space, expressed in terms of a change in percentage [126, 160]. In other words, one way of choosing such a reference point is to configure its (objective space) entries as being a fixed percentage above or below the bounds of the objective functions being maximised or minimised, respectively.

In order to illustrate the last two challenges mentioned above, reconsider the bi-objective optimisation problem example associated with Figure 8.1, and assume that the entries in objective space have been normalised. Then, the vector $\tilde{\boldsymbol{z}}$ shown in Figure 8.2 demonstrates a poor choice of reference point, as it is (visually) apparent that the hypervolume is significantly more sensitive to changes in values of the first objective function (f_1) than those of the second objective function (f_2).

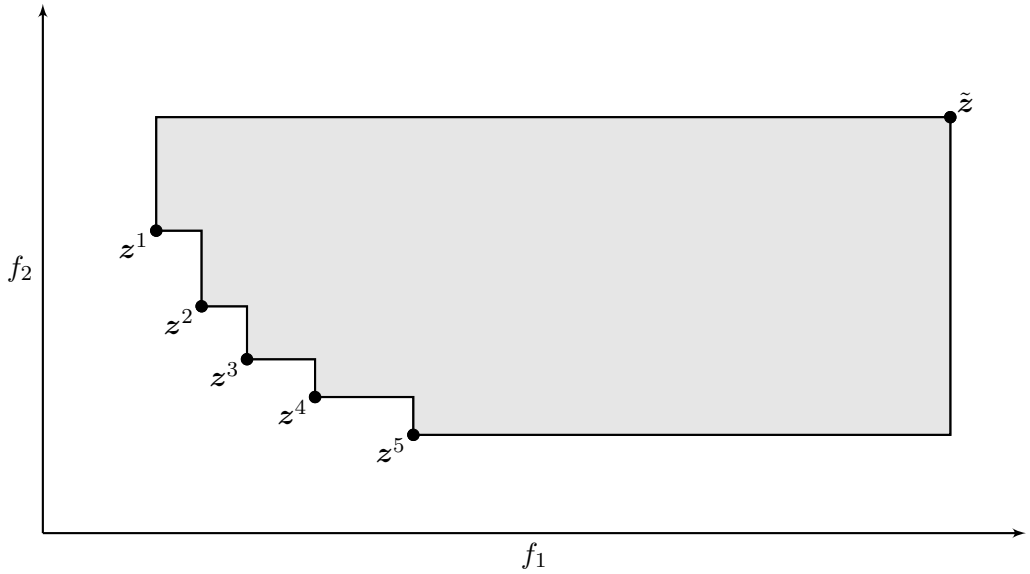


FIGURE 8.2: *A poor choice of reference point, $\tilde{\boldsymbol{z}}$, provided that the points in $\mathcal{S}_{\boldsymbol{Z}}$ have been normalised.*

Alternative uses

In addition to evaluating the performances of multiobective optimisation techniques, the hypervolume indicator may also be employed as a termination mechanism in population-based metaheuristics, where the search process is configured to end as soon as the change in hypervolume from the non-dominated front to the next is lower than a certain threshold value [142]. Of course, the smaller this threshold value is, the more iterations are usually required before termination of the algorithm. Furthermore, although one cannot guarantee the exact nature of the true Pareto front, it is also usually possible to establish some form of proximity interval between itself and the current non-dominated front in order to establish a guideline for such a stopping criterion.

The hypervolume indicator may also be used as a filtering mechanism in the sense that solutions in a non-dominated front are ranked according to their contribution towards the total hypervolume, so that the least contributing solutions may be removed from the front prior to

being presented to the decision maker. Moreover, integrating a diversity indicator of solutions spanning the non-dominated front into the hypervolume measure may be used as a means of finding highly-diversified non-dominated solution sets [143, 154].

8.1.2 The HSO algorithm

In this dissertation, the so-called *Hypervolume by Slicing Objectives* (HSO) algorithm was chosen as a means of comparing the performances of non-dominated fronts generated during experimental runs. This algorithm was proposed by While *et al.* [160] in 2006, and has mathematically been demonstrated to be the fastest known algorithm for computing hypervolume exactly at the time. Unlike other hypervolume indicators, such as the *Lebesgue measure*, in which non-dominated solutions are considered one at a time in respect of their contribution toward hypervolume performance, the HSO algorithm rather operates by processing the objectives one at a time. The basic working of the HSO algorithm is briefly described in this section. The interested reader is referred to the above-mentioned publication for a more thorough description and analysis of this algorithm.

The algorithm is initiated by sorting a single list $\mathcal{S}$ of N values in the non-dominated front, ranking these values with respect to the first of M objectives. These ranked values are used to cut cross-sectional *slices* through the objective space by expanding $\mathcal{S}$ into a set containing N lists of points in $M-1$ objectives and each of these lists is paired with the depth of the corresponding slice in the first objective[2]. Each slice is consequently an $(N-1)$-hypervolume in the remaining objectives at this stage, where the top slice contains only the point with the highest/lowest value in the first objective, the second slice contains the first two highest/lowest points, and so on, while the last slice contains all N points.

As explained in §8.1.1, dominated solutions do not contribute towards the hypervolume of a non-dominated front. And so, after reducing the points in each slice by one objective, some of these points may be dominated with respect to the remaining $M-1$ objectives. It follows that none of these dominated points will contribute any volume towards that slice and, consequently, newly dominated points uncovered within each slice are removed from it.

In the second step, the list of points in each of the N slices are sorted by ranking them with respect to the second objective[3], and these values are, once again, used to cut (sub)slices through the objective space by expanding each original slice into sets in $M-2$ objectives, so that each point in these sets are paired with the product of their respective depths in the first two objectives.

This process is continued. After each slicing action, the number of objectives is reduced by one, the newly dominated points within each slice are removed, and the remaining point within each slice are resorted according to the next objective. The process ends once the expansion has occurred $M-1$ times, resulting in a large number of slices, each containing one (non-dominated) point along the M^{th} objective (*i.e.* the one-objective hypervolume of that slice), each paired with the product of its depths in the first $M-1$ objectives. The depths of these objectives are thus accumulated multiplicatively throughout the algorithm. It follows that the original list of non-dominated points in M objectives is, in the end, reduced to a large number of singleton points, each paired with the cumulative depths of their M respective objective scores.

[2]The bounds provided on the one side by the corresponding estimated extremal value and, on the other side, by the corresponding reference point entry, need to be available in order to compute these depths.

[3]The computational performance of the HSO algorithm may vary significantly based on the order in which it processes the objectives in a given non-dominated front; as a result, certain studies attempt to identify a good order for processing the objectives along such a given front (*e.g.* see [159]).

The hypervolume is finally calculated by summing together the paired values associated with all these uni-dimensional slices.

8.1.3 Other performance measures

Perhaps the most natural way of comparing any two Pareto front approximation sets (*i.e.* non-dominated sets), generated by two different optimisation techniques, is to use the so-called *underlying preference structure* approach [58]. Similar to comparing any two solutions in the objective space of a multiobjective optimisation problem using the notion of dominance, three elementary outcomes are possible in such a comparison of two non-dominated sets $\mathcal{D}_1$ and $\mathcal{D}_2$: (1) either $\mathcal{D}_1$ is *better* than $\mathcal{D}_2$ (denoted by $\mathcal{D}_1 \rhd \mathcal{D}_2$), in which case every solution in $\mathcal{D}_2$ is weakly dominated by at least one solution in $\mathcal{D}_1$; or (2) $\mathcal{D}_2$ is better than $\mathcal{D}_1$; or (3) $\mathcal{D}_1$ and $\mathcal{D}_2$ are *incomparable* (denoted by $\mathcal{D}_1 || \mathcal{D}_2$), in which case neither front weakly dominates the other. It is, of course, always the case that $\mathcal{P} \rhd \tilde{\mathcal{P}}$ for any non-dominated set of solutions $\tilde{\mathcal{P}}$, where $\mathcal{P}$ denotes the Pareto optimal set of solutions. A case of incomparability between two non-dominated sets is illustrated in Figure 8.3 for a bi-objective minimisation problem with objective functions f_1 and f_2.

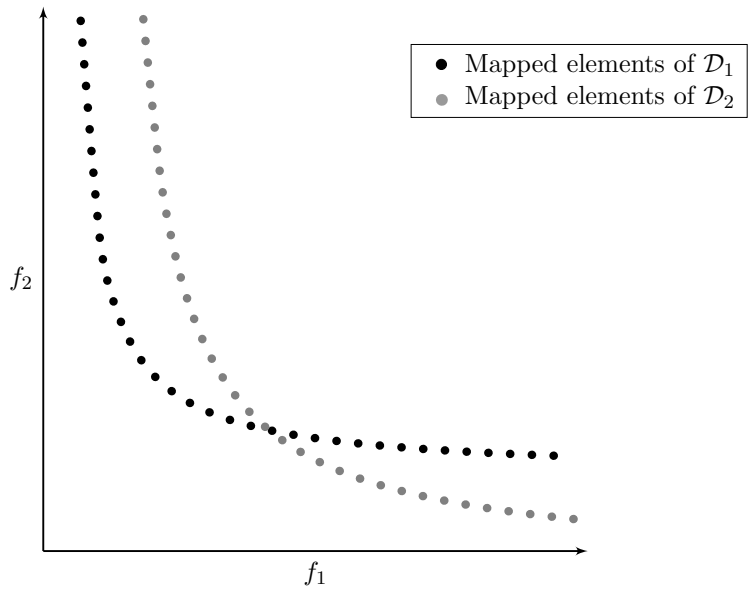


FIGURE 8.3: *Illustration of the mappings of two incomparable non-dominated sets $\mathcal{D}_1$ and $\mathcal{D}_2$ for a bi-objective minimisation problem.*

The types of statements listed above are, however, not conclusive in the sense of *how much* better one Pareto front approximation set is than another. The decision maker should rather be interested in quantifying their difference in quality on a continuous scale as well as investigating aspects such as the robustness and diversity of non-dominated solutions.

The method of *dominance ranking* may nevertheless be employed in assessing performance levels using this basic preference structure [58]. Here, the non-dominated sets obtained from multiple runs of several optimisation techniques[4] are combined into a *collection set* $\boldsymbol{C}$, so that each element in this set represents one of these non-dominated sets. Then, using the underlying preference structure for comparing sets in a partially ordered manner (as discussed above), it

[4]Or from the same optimisation techniques, but using various combinations of parametric configurations.

follows that any two elements in $\boldsymbol{C}$ are either incomparable or that one is better than the other. A rank may then be assigned to each element in $\boldsymbol{C}$ by, for example, simply counting the number of other elements that are better than it. That is,

$$\mathcal{D}^i_{rank} = 1 + \left|\{\mathcal{D}^j \rhd \mathcal{D}^i : i, j \in |\boldsymbol{C}|,\ i \neq j\}\right|.$$

A statistical analysis may then be conducted in order to evaluate the hypothesis that there exists a significant difference between the performance of the tested optimisation techniques. Dominance ranking is simple and straight-forward to implement. However, as mentioned earlier, performance assessment solely using partial ordering only provides very general statements to the decision maker; the degree to which one technique outperforms another is, for example, unknown in such an approach, and is moreover considered fairly independent of the decision maker's preference information.

In the case of stochastic optimisers, the notion of an *attainment function* may also be used [58]. The fundamental idea behind this concept speculates that the result of a stochastic optimisation process may be modelled as a probability distribution, described by a set $\boldsymbol{Y}$ of random objective function vectors. The attainment function is a reflection of this probability distribution and is based on the concept of *goal attainment*[5]. Informally speaking, this function may be described as a form of mean statistic of the set $\boldsymbol{Y}$.

A more similar technique to that of the hypervolume indicator is the class of *unary epsilon indicators*, which is also a type of quality indicator used in assessing the performances of multiobjective optimisation techniques. For descriptive purposes, consider a multiobjective optimisation problem with M minimisation objectives that have been normalised. Then, the epsilon indicator measure is defined as

$$I_\epsilon(\mathcal{D}, \boldsymbol{r}) = \inf_{\epsilon \in \mathbb{R}} \{\boldsymbol{x} \in \mathcal{D} : \boldsymbol{x} \succeq_\epsilon \boldsymbol{f}^{-1}(\boldsymbol{r})\},$$

where $\mathcal{D}$ is a non-dominated set of solutions, $\boldsymbol{r}$ is the chosen reference point and $\boldsymbol{x}_1 \succeq_\epsilon \boldsymbol{x}_2$ if and only if $\boldsymbol{x}_1|_o \leq \boldsymbol{x}_2|_o + \epsilon$ for all $o \in \{1, \ldots, M\}$, where $\boldsymbol{x}|_o$ denotes the objective function score of solution $\boldsymbol{x}$ with respect to objective o. The interested reader is referred to [58] for a more detailed description of the above-mentioned indicators.

8.2 On the growth of problem complexity

Prior to tackling the various system validation experimental procedures, it may be useful to consider the growth in model complexity of an MLE response selection problem domain. As with classical VRPs, the size of an MLE response selection problem domain is most essentially defined by the number of vehicles, the number of customers and the number of depots. Because the end-of-route assignment algorithm of §7.2 is applied to all search techniques in the experiments reported on in this dissertation, the number of bases may be ignored in this evaluation. Unlike standard VRPs, in which all customers are required to be visited, however, this problem needs to consider all possible subsets of customers to be visited. The aim in this section is therefore to derive a formula for estimating the exact size of an MLE response selection problem domain space for any given number of MLE resources and VOIs.

[5] An objective vector is said to be *attained* whenever it is weakly dominated by the non-dominated set generated during a run [58].

8.2.1 Permutation and combination coefficients

Permutations are better known as problems in which it is required to count the number of arrangements of a finite set of distinct objects or elements. It is well known [122] that the permutation count of K elements, $\Theta_p(K)$, is given by $K! := K(K-1)(K-2)\dots(3)(2)(1)$. Moreover, a so-called *k-permutation set* is defined as an ordered, non-repeating sequence of k elements, selected from a superset of K elements. The *permutation coefficient* $\Theta_p(K,k)$ is defined as the total number of ways in which a k-permutation may be formed from a set of K distinct elements. All k-permutations are therefore obtained by taking all possible k element subsets from a set of K elements and accumulating their respective permutation counts. Because the number of ways in which all K elements may be selected is equivalent to the number of ways in which k elements may be selected, taking into account the number of ways in which the remaining $K-k$ elements may be selected, it follows that $\Theta_p(K) = \Theta_p(K,k)\Theta_p(K-k)$, and so

$$\Theta_p(K,k) = \frac{\Theta_p(K)}{\Theta_p(K-k)} = \frac{K!}{(K-k)!}, \qquad k \leq K.$$

Furthermore, the number of ways in which k distinct elements may be selected from a finite set of K elements where the order of arrangement of these elements is not considered important is defined as the *combination coefficient* $\Theta_c(K,k)$. This coefficient may be derived by ignoring the ordinal condition imposed on the subset of elements selected, but rather only considering *which* elements are chosen. Because the number of permutations of k elements from K is the number of distinct subsets of k elements selected from K multiplied by the number of ways to order these respective subsets, the permutation coefficient may be reformulated as $\Theta_p(K,k) = \Theta_c(K,k)\Theta_p(k,k)$, and so it follows that

$$\Theta_c(K,k) = \frac{\Theta_p(K,k)}{\Theta_p(k,k)} = \frac{K!}{k!(K-k)!} =: \binom{K}{k}, \qquad k \leq K.$$

8.2.2 k-multicombinations

Consider a subset of size k selected from a set of K *indistinguishable* elements, and suppose that the elements of this subset must be associated with m *distinguishable* sets. The numbers of elements in these sets are denoted here as $k_1, \dots, k_m$. The number of multisubsets of size k, or *k-multicombinations* [93], is then the number of solutions to the equation

$$\sum_{i=1}^{m} k_i = k, \qquad k \leq K,\ k_i \in \{\mathbb{N} \cup \{0\}\},\ i \in \{1, \dots, m\}.$$

An isomorphism which may be employed to count the number of solutions to the above equation consists in counting the total number of ways in which k identical objects and $m-1$ identical dividers may be placed (uni-dimensionally) alongside one another. Because these dividers are identical, this process is equivalent to counting the number of ways in which k identical objects may be placed in $k+m-1$ distinct, empty slots (distinguished by the order in which they appear), with the dividers delimiting the slots. The number of solutions to this counting problem is therefore given by

$$\binom{k+m-1}{k}.$$

8.2.3 The permutated subgrouped k-multicombinations counting problem

Consider a set of K *distinguishable* objects and $m-1$ dividers producing m *distinguishable* compartments. But suppose this time that it is required to count the number of ways in which a subset of k objects, selected from the set of K objects, may be inserted into the box compartments, assuming that there is no maximum capacity constraint on the number of objects allocated to any compartment and that some compartments may remain empty. This is easily assessed by applying the so-called *product rule* [122], where each object has m possible allocation options, independently of the allocation statuses of the remaining objects. In other words, there are m^k possible ways in which k distinct objects may be partitioned into m distinct compartments.

Ultimately, it is required to count the number of ways in which these objects may be distributed amongst the compartments in such a way that the order in which the subsets of objects are arranged amongst all compartments matters. That is, once the k objects have been allocated to the $k+m-1$ slots, they must be reordered distinctly within these respective slots in order to produce different permutations. This counting problem may be solved by employing the isomorphism described in §8.2.2. The total number of solutions to this enumeration problem is therefore given by

$$k!\binom{k+m-1}{k}, \qquad k \leq K.$$

It therefore follows from §8.2.1 that

$$\sum_{k=0}^{K} k!\binom{K}{k}\binom{k+m-1}{k} = \sum_{k=0}^{K}\binom{K}{k}\frac{(k+m-1)!}{(m-1)!} \tag{8.1}$$

is the enumeration of all possible subsets of k distinct objects over all possible sizes of $k \in \{0,1,\ldots,K\}$ assigned to m distinct compartments, characterised by $m-1$ distinct dividers, in which the overall permutation order matters.

This result is referred to here as the *permutated subgrouped k-multicombinations* counting problem, which is analogous to counting the exact number of solutions spanning the domain space of an MLE response selection problem instance in which there are m available MLE resources (the distinct compartments) and K VOIs (the distinct objects). In order to gauge the problem size as a function of m and K according to (8.1), numerical counts for various combinations of m and K are shown in Table 8.1.

8.2.4 Discussion

It is clear, based on the results in Table 8.1, that the complexity of the MLE response selection problem increases rapidly with increases of the number of VOIs and MLE resources at hand. Although such domain sizes may seem overwhelming, particularly in view of limited time available to find good solutions to a given problem instance, tools such as model management features and intelligent exploration moves are expected to condense the actual decision space that has to be considered explicitly, referred to here as the search space, into a small portion of the entire domain[6]. It is, moreover, acknowledged that the proportion of feasible solutions residing within a given search space is also expected to be relatively small. A visual representation of these sets

[6] A good upper bound on the size of an MLE response selection search space may be obtained by assuming that, practically speaking, an MLE resource will most often never be assigned more than a certain number of VOIs as it is not necessary to plan too far into the future due to various solution progression uncertainty factors (see Chapter 6). But this is analogous to branching the permutated subgrouped k-multicombinations counting

	m									
K	1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10	11
2	5	11	19	29	41	55	71	89	109	131
3	16	49	106	193	316	481	694	961	1 288	1 681
4	65	261	685	1 457	2 721	4 645	7 421	11 265	16 417	23 141
5	326	1 631	5 056	12 341	25 946	49 171	86 276	142 601	224 686	340 391
6	1 957	11 743	42 079	116 125	271 801	566 827	1 084 483	1 940 089	3 288 205	5 330 551
7	13 700	95 901	390 454	1 203 329	3 105 936	7 073 725	14 665 106	28 245 729	51 263 164	88 577 021
8	109 601	876 809	4 000 441	13 627 073	38 474 561	95 064 361	2.1×10^8	4.4×10^8	8.5×10^8	1.6×10^9
9	986 410	8 877 691	44 881 660	1.7×10^8	5.1×10^8	1.4×10^9	3.3×10^9	7.2×10^9	1.5×10^{10}	2.9×10^{10}
10	9 864 101	98 641 011	5.5×10^8	2.2×10^9	7.4×10^9	2.1×10^{10}	5.4×10^{10}	1.3×10^{11}	2.7×10^{11}	5.6×10^{11}
11	68 588 311	7.1×10^8	4.1×10^9	1.71×10^{10}	5.81×10^{10}	1.7×10^{11}	4.43×10^{11}	1.05×10^{12}	2.33×10^{12}	4.84×10^{12}

TABLE 8.1: *Values of the expression in (8.1) for* $m \in \{1, \ldots, 10\}$ *and* $K \in \{1, \ldots, 11\}$.

is shown in Figure 8.4. Lastly, the numerical experiments performed later in this chapter reveal that an MLE response selection feasible search space may actually be chaotic (see §2.2), fragmented and possibly disjoint, rather than "smooth" and "predictable." An artist's impression of such a search space is also included in Figure 8.4 [84].

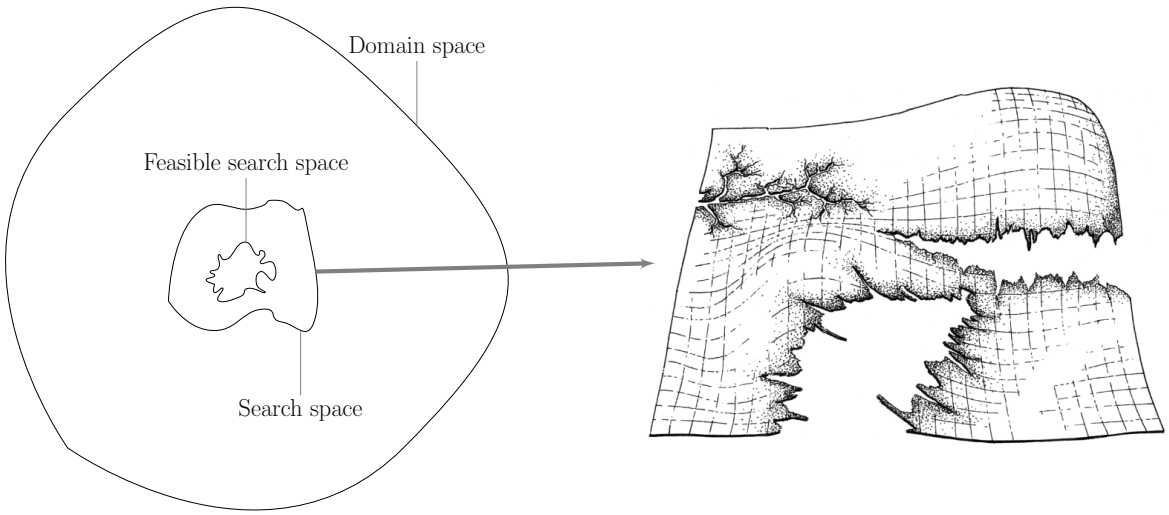


FIGURE 8.4: *Representations of the domain space, search space and feasible search space of a hypothetical combinatorial optimisation problem similar to one studied in this dissertation.*

8.3 A lower-complexity hypothetical scenario

In order to demonstrate the working of the proposed mathematical modelling subsystem and optimisation methodology component, described in Chapters 4, 5 and 7, a hypothetical, lower-complexity deterministic MLE response selection scenario within an intermediate decision making paradigm is put forward in this section. The proposed scenario mimics a situation at sea in which eleven VOIs have been detected and evaluated, after which a decision must be made in respect of the allocation of MLE resources to these VOIs. Here, the external MLE resource assignment infrastructure consists of eight MLE resources, four coastal bases, seven patrol circuits and three decision making entities. It is assumed that the jurisdiction area is defined analogously to the EEZ, and that threat detection and threat evaluation input data are available in the MLE database at the start of the current hypothetical time stage, and that all relevant parameter values and model management input data have been established. The various parameters relevant to this scenario together with a formulation of the mathematical model employed are presented in this section.

8.3.1 Fixed input parameters

In the scenario, eight known threat types are associated with the MLE environment of the hypothetical coastal nation. In addition, and as mentioned in Chapter 4, the threat evaluation

problem of §8.2.3 into a contingency formulation where each slot may never be assigned more than a certain number of objects. Although not pursued in this dissertation, it is suggested that this ancillary counting problem may be solved with the use of generating functions.

Threat type	1	2	3	4	5	6	7	8	9	10
Threat Intensity	0.3	0.6	0.3	0.7	0.3	0.9	0.8	1	0.5	0

TABLE 8.2: *Threat type intensity levels as judged by the hypothetical coastal nation.*

process must also account for the possibility of an unknown threat class and a false alarm class. The proposed threat categories are therefore indexed by the set $\mathcal{H} = \{1, \ldots, 10\}$, and the relative priority levels[7] assigned by the coastal nation in respect of neutralising each of these threat categories are listed in Table 8.2.

In addition to a centralised operator, three decision making entities are assumed to be responsible for carrying out the MLE response selection operations of the hypothetical coastal nation, referred to as *Entity 1*, *Entity 2* and *Entity 3*, and indexed by the set $\mathcal{Z} = \{1, 2, 3\}$. The relative importance values[7] of each of these entities to the coastal nation is specified by the values $Z_1 = 1$, $Z_2 = 0.8$ and $Z_3 = 0.5$. The general purpose of Entity 1 is therefore assumed to bear a higher importance than that of Entities 2 and 3 to the coastal nation. At the start of the scenario, it is assumed that the hypothetical coastal nation possesses eight available MLE resources indexed by the set $\mathcal{V}^r = \{a, \ldots, h\}$. Here, MLE resources a and b are assumed to belong to Entity 1, while MLE resources c and d belong to Entity 2. The remaining MLE resources are assumed to belong to Entity 3.

In §4.1.6 it was speculated that, as a result of their unique infrastructures, sizes, crew expertises or speeds, certain types of MLE resources excel, or do not perform well, with respect to countering VOIs embodying certain types of threats. More importantly, the notion of infeasible encounters was introduced. Recall that such an encounter takes place whenever an MLE resource is incapable of neutralising the type of threat embodied by a VOI assigned to it for interception. Three modelling tools were provided for lowering the risk of infeasible encounters. One of these tools involved the introduction of an additional objective to the mathematical model (see Objective IV in §4.2.3). The other tools consist of making use of VOI exclusion sets and employing stochastic threshold parameters, as described in §5.2.1 and §6.6, respectively. Additionally, a simpler method to discourage such assignments may involve the use of soft constraints for penalising the visitation score (see Objective I in §4.2.3) accordingly, by assigning relatively large negative values to infeasible encounter parameters, as well as penalising the delay score (see Objective II in §4.2.3), by specifying very large expected service times for such encounters. For the sake of simplicity, the latter approach is employed in this chapter.

The scores associated with the effectiveness of each MLE resource in terms of neutralising each type of threat are listed in Table 8.3, where M^- is some arbitrarily chosen (relatively large) negative real number reserved for use in lowering the risk of infeasible encounters. In addition, the estimated service times associated with the MLE resources and the various types of threats are also displayed in the table, where M^+ is some arbitrarily chosen positive (relatively large) real number. Note that the expected service time taken by an MLE resource to neutralise a certain threat type is assumed to have an inversely proportional relationship with respect to its efficiency at neutralising threats of that type. As with counter-threat performances, MLE resources that are strictly incapable of neutralising certain types of threats are, as explained above, assigned a large service time for these types of threats.

The MLE boundaries of the hypothetical coastal nation in this scenario are designed as an interpolated curve passing through the ordered set of coordinate pairs

$$\{(-500, 0), (-350, 280), (-40, 400), (235, 330), (500, 0)\}$$

[7]Normalised to values within the real unit interval.

Threat type	MLE resource							
	a	b	c	d	e	f	g	h
1	(0.4,0.6)	(0.4,0.6)	(0.3,0.7)	(0.3,0.7)	(1,0.1)	(1,0.1)	(1,0.1)	(1,0.1)
2	(0.7,0.3)	(0.8,0.2)	(0.7,0.3)	(0.7,0.3)	(0.3,0.7)	(0.3,0.7)	(M^-,M^+)	(M^-,M^+)
3	(0.3,0.7)	(0.3,0.7)	(0.9,0.1)	(0.9,0.1)	(0.6,0.4)	(0.6,0.4)	(0.4,0.6)	(0.4,0.6)
4	(0.2,0.8)	(0.2,0.8)	(0.9,0.1)	(0.9,0.1)	(0.5,0.5)	(0.5,0.5)	(0.3,0.7)	(0.3,0.7)
5	(1,0.3)	(0.8,0.2)	(0.8,0.2)	(0.8,0.2)	(0.6,0.4)	(0.6,0.4)	(0.3,0.7)	(0.3,0.7)
6	(1,0.5)	(0.9,0.45)	(0.3,0.7)	(0.3,0.7)	(M^-,M^+)	(M^-,M^+)	(M^-,M^+)	(M^-,M^+)
7	(1,0.5)	(0.9,0.45)	(0.3,0.7)	(0.3,0.7)	(M^-,M^+)	(M^-,M^+)	(M^-,M^+)	(M^-,M^+)
8	(1,0.5)	(0.9,0.45)	(M^-,M^+)	(M^-,M^+)	(M^-,M^+)	(M^-,M^+)	(M^-,M^+)	(M^-,M^+)
9	(0.8,0.3)	(0.6,0.3)	(0.5,0.4)	(0.5,0.4)	(0.4,0.5)	(0.4,0.5)	(M^-,M^+)	(M^-,M^+)
10	(1,0.1)	(1,0.1)	(1,0.1)	(1,0.1)	(1,0.1)	(1,0.1)	(1,0.1)	(1,0.1)

TABLE 8.3: *MLE resource counter-threat ability levels, as first entries, and expected service times (hours), as the second entries.*

and the coastal boundary passing through the ordered set of coordinate pairs

$$\{(-250, 0), (-175, 110), (-40, 155), (110, 105), (250, 0)\},$$

as illustrated in Figure 8.5.

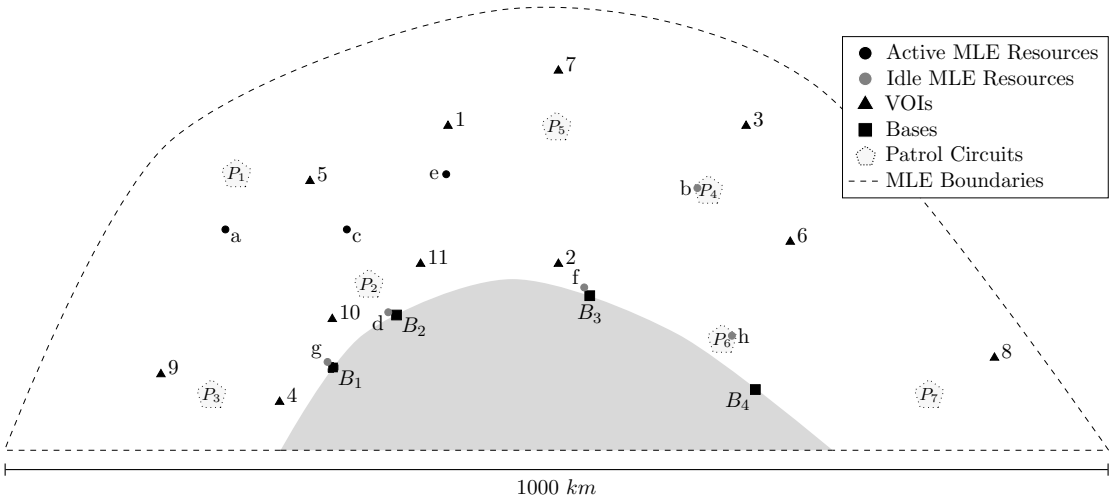


FIGURE 8.5: *Graphical representation of the physical elements in the lower-complexity hypothetical scenario.*

The four coastal bases in this scenario, labelled B_1, B_2, B_3 and B_4, and indexed by the set $\mathcal{V}^b = \{1, 2, 3, 4\}$, are found at the respective coordinate locations $(-202, 75)$, $(-145, 122)$, $(30, 140)$ and $(180, 55)$. In addition, seven patrol circuits, labelled $P_1, P_2, P_3, P_4, P_5, P_6$ and P_7, and indexed by the set $\mathcal{V}^p = \{1, \ldots, 7\}$, are located at the respective coordinate locations $(-290, 250)$, $(-170, 150)$, $(-312, 50)$, $(138, 235)$, $(0, 392)$, $(150, 100)$ and $(338, 50)$ within the jurisdiction area of the coastal nation.

Attribute information associated with the average speed, setup cost, setup time, travel cost, distance autonomy and time autonomy of each MLE resource may be found in Table 8.4. In addition, entries for the parameters $\beta_{bk} \in \{0, 1\}$ are provided at the bottom of the table, where (as defined in §4.2.1) $\beta_{bk} = 0$ implies that MLE resource k is prevented from ending its route at base b during any given time stage, or $\beta_{bk} = 1$ otherwise (independently from model management preferences in respect of end-of-route assignments, as discussed in §5.2.3). Furthermore,

information on the distance and time patrol autonomy thresholds associated with every MLE resource (as discussed in §4.1.7) is provided in Table 8.5.

Attribute	MLE resource							
	a	b	c	d	e	f	g	h
Average Speed (km/h)	40	50	50	50	70	70	60	60
Setup Costs ($)	4 000	3 500	800	800	500	500	200	200
Setup Time (h)	1	0.75	0.5	0.5	0.3	0.3	0.2	0.2
Travelling Costs $(\$\backslash km)$	40	35	20	20	15	15	5	5
Distance Autonomy (km)	3 500	2 500	700	700	500	500	400	400
Time Autonomy (h)	720	360	96	96	48	48	24	24
B_1 Allocation Status (β_{1k})	1	1	0	0	1	1	1	1
B_2 Allocation Status (β_{2k})	0	0	1	1	1	1	1	1
B_3 Allocation Status (β_{3k})	1	1	1	1	1	1	1	1
B_4 Allocation Status (β_{4k})	0	0	1	1	1	1	1	1

TABLE 8.4: *MLE resource attribute characteristics.*

MLE Resource	Patrol Circuit						
	P_1	P_2	P_3	P_4	P_5	P_6	P_7
a	(800,400)	(600,300)	(600,300)	(800,350)	(800,350)	(800,250)	(1 100,300)
b	(700,200)	(500,150)	(500,150)	(700,250)	(700,250)	(700,130)	(1 100,150)
c	(500,50)	(400,50)	(600,45)	(400,45)	(450,45)	(250,35)	(350,45)
d	(500,50)	(400,50)	(600,45)	(400,45)	(450,45)	(250,35)	(350,45)
e	(400,25)	(200,25)	(200,23)	(300,22)	(350,23)	(200,16)	(300,22)
f	(400,25)	(200,25)	(200,23)	(300,22)	(350,23)	(200,16)	(300,22)
g	(350,13)	(175,13)	(175,12)	(250,12)	(300,12)	(175,10)	(250,11)
h	(350,13)	(175,13)	(175,12)	(250,12)	(300,12)	(175,10)	(250,11)

TABLE 8.5: *Patrol circuit distance autonomy thresholds (km), as first entries, and patrol circuit time autonomy thresholds (hours), as second entries.*

8.3.2 Dynamic input parameters

At time $t = 0$ (the beginning of the time stage associated with the hypothetical problem instance), it is assumed that eleven VOIs, indexed by the set $V^e = \{1, \ldots, 11\}$, are tracked in the jurisdiction area of the coastal nation. Input data on the geographical locations, estimated trajectories, threat probability vectors and time elapsed since the detection of these VOIs, are displayed in Table 8.6.

The geographical locations of MLE resources at time $t = 0$, along with their distance and time autonomy levels at that time, as well as whether or not they are idle and not in patrol mode (as defined by the parameter $\gamma_{k\tau}$, as described in §4.2.1), are listed in the left half of Table 8.7. Note that the autonomy levels of idle MLE resources located at bases are at their maximum levels, thus matching the corresponding autonomy levels in Table 8.4. In addition, model management input data derived from the subjective expertise of the (central) MLE response selection operator and idle MLE resources management DSS with respect to end-of-route assignments, VOI inclusion sets[8] and VOI exclusion sets, are displayed in the right half of Table 8.7.

[8]In this scenario, unordered VOI inclusion sets are assumed to be implemented.

VOI	Location	Estimated itinerary $(x(t), y(t))$	Threat probabilities	Delay (hours)
1	$(-100, 300)$	$(-100, 300)$	$(0, 0, 0.7, 0.2, 0, 0, 0, 0, 0, 0.1)$	1.5
2	$(0, 175)$	$(-5t, 175 - 10t)$	$(0.7, 0, 0, 0.1, 0, 0, 0, 0, 0, 0.2)$	0.5
3	$(170, 300)$	$(170 - 5t, 300 - 10t)$	$(0, 0, 0, 0, 0, 0.35, 0.3, 0, 0.3, 0.05)$	2
4	$(-250, 50)$	$(-250 - 10t, 50 - 5t)$	$(0.7, 0, 0, 0, 0, 0, 0, 0, 0, 0.3)$	0.25
5	$(-200, 250)$	$(-200 - 15t, 250)$	$(0, 0, 0, 0, 0.7, 0, 0, 0.2, 0, 0.1)$	1
6	$(210, 195)$	$(210 - 5t, 195 + 15t)$	$(0, 0.7, 0, 0, 0.3, 0, 0, 0, 0, 0)$	0.5
7	$(0, 350)$	$(-25t, 350)$	$(0, 0, 0, 0, 0, 0, 0, 0.75, 0, 0.25)$	2
8	$(395, 90)$	$(395, 90 + 10t)$	$(0, 0, 0, 0, 0.5, 0, 0, 0, 0.4, 0.1)$	1.75
9	$(-360, 75)$	$(-360, 75 + 20t)$	$(0, 0, 0, 0, 0, 0, 0.6, 0.4, 0, 0)$	0.45
10	$(-200, 125)$	$(-200, 125)$	$(0, 0, 0.3, 0.6, 0, 0, 0, 0, 0, 0.1)$	0.25
11	$(-120, 175)$	$(-120 + 5t, 175 + 10t)$	$(0.8, 0, 0, 0, 0, 0, 0, 0, 0, 0.2)$	0.5

TABLE 8.6: *Input data for detected VOIs.*

MLE resource	Current location	Distance autonomy (kms)	Time autonomy $(hours)$	γ_k	Base exclusion sets	Patrol circuit inclusion sets	VOI exclusion sets	VOI inclusion sets
a	$(-300, 200)$	3000	500	0	$\emptyset$	$\{P1, P_3, P_5\}$	$\emptyset$	$\emptyset$
b	$(128, 125)$	1800	220	0	$\emptyset$	$\{P4, P_5, P_7\}$	$\emptyset$	$\{7\}$
c	$(-150, 200)$	650	95	0	$\{B_2\}$	$\{P1, P_2, P_3, P_5\}$	$\{9\}$	$\emptyset$
d	$(-145, 122)$	700	96	1	$\emptyset$	$\{P2, P_4, P_5, P_6\}$	$\emptyset$	$\emptyset$
e	$(-100, 250)$	350	30	0	$\{B_1, B_2\}$	$\{P2, P_4, P_5\}$	$\emptyset$	$\emptyset$
f	$(30, 140)$	500	48	1	$\{B_1\}$	$\emptyset$	$\emptyset$	$\{2\}$
g	$(-202, 75)$	400	24	1	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
h	$(395, 90)$	325	21	0	$\{B_1, B_2, B_3\}$	$\emptyset$	$\emptyset$	$\emptyset$

TABLE 8.7: *Input data for MLE resources and model management.*

Finally, certain input data are assumed to have been specified by the decision making entities with respect to their preferences in dealing with the detected VOIs (as described in §4.3). These data are shown in Table 8.8. The physical elements of this hypothetical scenario at time $t = 0$ are all displayed in Figure 8.5.

8.3.3 Mathematical model

The hypothetical problem instance described above is solved during time stage $\tau \in \mathbb{N}$ in an intermediate MLE response selection decision making paradigm, as described in Chaper 3, involving Objectives I, II, III and V of Chapter 4. It is assumed that the costs associated with trajectory deviations may be ignored and that no MLE resources are currently in the process of servicing any observed VOIs.

Entity	VOI preference ordered set	Ideal quantity
1	$\mathcal{O}_1 = \{9, 7, 3, 5, 6, 8, 1, 10, 2, 11, 4\}$	$N_1 = 3$
2	$\mathcal{O}_2 = \{1, 3, 8, 10, 5, 7, 9, 6, 11, 4, 2\}$	$N_2 = 3$
3	$\mathcal{O}_3 = \{11, 4, 2, 1, 5, 8, 6, 10, 3, 9, 7\}$	$N_3 = 4$

TABLE 8.8: *Input data received from the decision making entities.*

The aim in this tetra-objective MLE response selection model is therefore to

$$\begin{aligned}
&\text{maximise} && \sum_{i\in\mathcal{V}_\tau^e}\sum_{k\in\mathcal{V}^r} y_{ik\tau}\sum_{h\in\mathcal{H}} Q_h W_{kh} p_{ih\tau},\\
&\text{minimise} && \sum_{i\in\mathcal{V}_\tau^e} t_{i\tau}\sum_{k\in\mathcal{V}^r} y_{ik\tau}\sum_{h\in\mathcal{H}} p_{ih\tau} Q_h,\\
&\text{minimise} && \sum_{k\in\mathcal{V}^r}\left(\gamma_{k\tau}C_k^s\sum_{j\in\mathcal{V}_\tau^e} x_{0_{k\tau}jk\tau} + \Gamma_k\sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}}\sum_{\substack{j\in\mathcal{V}_\tau\\ j\neq 0_{k\tau}}} x_{ijk\tau}d_{ijk\tau}\right), \text{ and}\\
&\text{maximise} && \sum_{s\in\mathcal{Z}} Z_s\left[\sum_{i\in\mathcal{V}_\tau^e} z_{is\tau}\left(n_\tau - \mathcal{O}_{s\tau}(i)\right) - f_c\left(N_{s\tau} - \sum_{i\in\mathcal{V}_\tau^e} z_{is\tau}\right)\right]
\end{aligned}$$

subject to the constraints

$$\begin{aligned}
\sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}}\sum_{\substack{j\in\mathcal{V}_\tau\\ j\neq 0_{k\tau}}} x_{ijk\tau} &= \sum_{\ell\in\mathcal{V}_\tau^e} y_{\ell k\tau}, && k\in\mathcal{V}^r,\\
\sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}} x_{ijk\tau} - \sum_{\substack{\ell\in\mathcal{V}_\tau\\ \ell\neq 0_{k\tau}}} x_{j\ell k\tau} &= 0, && j\in\mathcal{V}_\tau^e,\ k\in\mathcal{V}^r,\\
\sum_{i\in\mathcal{V}_\tau^e}\sum_{\substack{j\in\mathcal{V}_\tau^e\\ j\neq i}} x_{ijk\tau} &\leq |\mathcal{V}_{k\tau}^e| - 1, && k\in\mathcal{V}^r, |\mathcal{V}_{k\tau}^e|\geq 2,\\
\sum_{k\in\mathcal{V}^r} y_{ik\tau} &\leq 1, && i\in\mathcal{V}_\tau^e,\\
\sum_{s\in\mathcal{Z}} z_{is\tau} &\leq 1, && i\in\mathcal{V}_\tau^e,\\
\sum_{j\in\mathcal{V}_\tau^e}\sum_{k\in\mathcal{V}^r} x_{0_{k\tau}jk\tau} &= \sum_{i\in\mathcal{V}_\tau^e}\sum_{\substack{\ell\in\mathcal{V}^b\\ \ell\in\mathcal{V}^p}}\sum_{k\in\mathcal{V}^r} x_{i\ell k\tau}\\
\sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}}\sum_{\substack{j\in\mathcal{V}_\tau\\ j\neq 0_{k\tau}}} d_{ijk\tau}x_{ijk\tau} &\leq a_{k\tau}^d, && k\in\mathcal{V}^r,\\
\sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}_\tau^e}}\sum_{\substack{j\in\mathcal{V}_\tau\\ j\neq 0_{k\tau}}} \frac{d_{ijk\tau}}{\eta_k}x_{ijk\tau} &\leq a_{k\tau}^t, && k\in\mathcal{V}^r,\\
\sum_{i\in\mathcal{V}_\tau^e} x_{ibk} &\leq \beta_{bk}, && b\in\mathcal{V}^b,\ k\in\mathcal{V}^r,\\
-(a_{k\tau}^d - \tilde{A}_d - A_{k\rho}^d) &\leq A_{k\rho}^d(1-w_{k\rho\tau}^d), && k\in\mathcal{V}^r,\ \rho\in\mathcal{V}^p,\\
x_{\ell\rho k\tau} &\leq w_{k\rho\tau}^d, && \ell\in\mathcal{V}_\tau^e,\ k\in\mathcal{V}^r,\ \rho\in\mathcal{V}^p,\\
-(a_{k\tau}^t - \tilde{A}_t - A_{k\rho}^t) &\leq A_{k\rho}^t(1-w_{k\rho\tau}^t), && k\in\mathcal{V}^r,\ \rho\in\mathcal{V}^p,\\
x_{\ell\rho k} &\leq w_{k\rho\tau}^t, && \ell\in\mathcal{V}_\tau^e,\ k\in\mathcal{V}^r,\ \rho\in\mathcal{V}^p,
\end{aligned}$$

$$
\begin{aligned}
w^d_{k\rho\tau}, w^t_{k\rho\tau} &\in \{0,1\}, && k \in \mathcal{V}^r, \rho \in \mathcal{V}^p,\\
x_{ijk\tau} &\in \{0,1\}, && i \in \{0_{k\tau}\} \cup \mathcal{V}^e_\tau,\\
& && j \in \mathcal{V}^e_\tau \backslash \{0_{k\tau}\},\ k \in \mathcal{V}^r,\\
y_{ik\tau} &\in \{0,1\}, && i \in \mathcal{V}^e_\tau,\ k \in \mathcal{V}^r,\\
z_{is\tau} &\in \{0,1\}, && i \in \mathcal{V}^e_\tau,\ s \in \mathcal{Z},
\end{aligned}
$$

where

$$\tilde{A}_d = \sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}^e_\tau}} \sum_{\substack{j\in\mathcal{V}^e_\tau\\ j\neq 0_{k\tau}}} (d_{ijk\tau} x_{ijk\tau}) - d_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau} x_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau}$$

and

$$\tilde{A}_t = \sum_{\substack{i=0_{k\tau}\\ i\in\mathcal{V}^e_\tau}} \sum_{\substack{j\in\mathcal{V}^e_\tau\\ j\neq 0_{k\tau}}} \left(\frac{d_{ijk\tau}}{\eta_k} x_{ijk\tau}\right) - \frac{d_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau}}{\eta_k} x_{\mathcal{V}^e_{k\tau}(|\mathcal{V}^e_{k\tau}|)\rho k\tau}.$$

8.4 Optimisation procedures and results

The experimental results obtained by solving the hypothetical MLE response selection problem instance of §8.3 (using the optimisation algorithms put forward in Chapter 7) are presented in this section and their performances are analysed. The main priority in the interpretation of these results, however, rather evolves around identifying combinations of parameter values that may be used in the design of the search process of a real-life MLE response selection DSS, for solving problems of similar complexities and nature as that configured in this scenario. As discussed previously, because computation time budget is a critical factor when solving instances of the MLE response selection problem, a time limit was employed as the stopping criterion for all metaheuristic search techniques.

At first it seems that choosing the correct optimisation performance measures is not an obvious task. This is because more information has to be acquired in order to reflect an MLE response selection decision maker's preferences and values in a real-life environment. Due to its numerous benefits, the hypervolume indicator described in §8.1.1, computed by the HSO algorithm, was, however, selected as a means of testing the performances of the proposed optimisation techniques. It is therefore assumed that this indicator reflects the preferences and values of the decision maker. Implementation of the alternative performance measure techniques described in §8.1.3 is not pursued in this dissertation.

It was decided that all VOIs belonging to VOI inclusion sets are to be removed from a solution string prior to performing any general solution transformation procedure upon it. The aim in doing this, as explained before, is to reduce the computational waste associated with the excessive production of infeasible solutions uncovered during a search process by excluding VOIs belonging to these sets from taking part in exploration transformation procedures.

Because of the level of uncertainty associated with the functionality of certain parameters in each optimisation technique, a series of preliminary experiments were conducted for both the method of simulated annealing and the NSGA-II in order to gain a better understanding of the functionality of these parameters (particularly in respect of their sensitivities towards hypervolume contribution). The interested reader is referred to Tables A.1–A.4 in Appendix A for a review of the results of these experiments. Each experiment in these tables consisted of five runs

(thus generating five non-dominated fronts in each case) subject to both 1- and 2-minute computational budgets. It should be noted that many more experiments had to be conducted using the proposed method of simulated annealing (namely 63) due to the relatively large number of search parameters associated with this method (in contrast to 24 experiments involving the NSGA-II). Furthermore, for performance assessment purposes, statistics for the average number of iterations/generations ($\overline{I}$ and $\overline{G}$, respectively), the average size of the non-dominated front ($|\overline{\mathcal{A}}|$ for the method of simulated annealing and $|\overline{\mathcal{F}_1}|$ for the NSGA-II) and the average hypervolume quantity $\overline{H}_{vol}$ were recorded for each experiment. In the method of simulated annealing, the number of iterations refers to the *successful* number of transformations performed on a solution throughout the algorithm execution (that is, the total number of feasible neighbour solutions generated), not to the number of times a solution transformation was applied (irregardless of the outcome) or to the number of times feasible neighbouring solutions were accepted according to the Metropolis rule of acceptance.

Based on the output analysis of these preliminary experiments, further robust experiments were conducted employing well-performing combinations of parametric configurations. These parameter value combinations were based on the results in the tables of Appendix A, an understanding of the nuances associated with the metaheuristics at hand (as discussed in Chapter 2) and subjective choices. Each of these more robust experiments also consisted of five runs each, but subject to 2- and 5-minute computational budgets instead. The results obtained from any two experiments within the same termination criterion time frame were compared to one another by recording their average hypervolume measures (which are to be maximised), as well as their hypervolume standard deviations, $s(\overline{H}_{vol})$ (which are to be minimised). Here, given a high average hypervolume associated with any given experiment, a smaller standard deviation around the expected performance level for the respective search technique and associated parameter value combination is, of course, more desirable (particularly for risk-averse MLE response selection decision makers), as it builds up the operator's confidence in trusting the DSS to produce high-quality trade-off alternatives to an MLE response selection problem instance every time such a specifically configured search technique is employed.

The computations required in all experiments were performed on personal computers with 3.0 GHz Intel®Core^{TM} 2 Duo E8400 processors and 3.25 GB RAM, running in Ubuntu Gnome 3.4.2. The experiments were configured and performed in the software package MATLAB [147]. Due to the nature of the stopping criterion employed in these experiments, care was taken to ensure that each run was allocated the same processing power for the duration of the search processes (*i.e.* the processing power of a computer was not allocated to perform any other demanding tasks on the same machine).

8.4.1 Objectives normalisation and selection of a reference point

Adopting a reverse engineering approach, numerical bounds associated with each objective function in this scenario were closely approximated. These bounds are shown in Table 8.9. It is noted that the lower bounds for Objectives II and III are not zero as the null vector in this scenario is an infeasible solution.

In order not to favour certain objective functions over others in respect of hypervolume contribution, the points in each non-dominated front were normalised according to the function

$$f^1_{norm} = \frac{f(\boldsymbol{x})|1 - f^1_{min}}{4.8 - f^1_{min}}$$

along the first objective, where $f(\boldsymbol{x})|1$ is the visitation score associated with solution $\boldsymbol{x}$ and f^1_{min}

Objective	Type of bound	Numerical bound
Visitation score	upper bound	4.8
Total delay	lower bound	7.75 (hours)
Operating costs	lower bound	\$5 600
Consensus level	upper bound	91

TABLE 8.9: *Approximate evaluation of objective function bounds for the lower-complexity hypothetical scenario.*

is the lowest visitation score recorded in the non-dominated set, the function

$$f^2_{norm} = \frac{f(\boldsymbol{x})|2 - 7.75}{f^2_{max} - 7.75}$$

along the second objective, where $f(\boldsymbol{x})|2$ is the total delay associated with solution $\boldsymbol{x}$ and f^2_{max} is the highest total delay recorded in the non-dominated set, the function

$$f^3_{norm} = \frac{f(\boldsymbol{x})|3 - 5\ 600}{f^3_{max} - 5\ 600}$$

along the third objective, where $f(\boldsymbol{x})|3$ is the operating cost associated with solution $\boldsymbol{x}$ and f^3_{max} is the highest operating cost recorded in the non-dominated set, and the function

$$f^4_{norm} = \frac{f(\boldsymbol{x})|4 - f^4_{min}}{91 - f^4_{min}}$$

along the fourth objective, where $f(\boldsymbol{x})|4$ is the consensus level associated with solution $\boldsymbol{x}$ and f^4_{max} is the lowest consensus level recorded in the non-dominated set. The reference point was chosen as the vector $\boldsymbol{r} = (-0.05, 1.05, 1.05, -0.05)$.

8.4.2 Simulated annealing results

The experiments performed using the multiobjective simulated annealing algorithm proposed in §7.5 employ four different solution transformation techniques triggered at every iteration of the algorithm according to a certain probability distribution. These transformations are a *within-route swap* transformation with associated probability p_{wr}, a *between-route swap* transformation with associated probability p_{brs}, a *between-route delete-and-insert* transformation with associated probability p_{brd}, and an *inter-state* transformation with associated probability p_{is}. Descriptions of the above solution transformations may be found in §7.5.3. In addition, the exponential growth distribution approach for inter-state transformations was adopted, and so the parameters (α, β) of §7.5.4 also had to be specified.

The well-known and widely used *geometric* temperature cooling schedule[9] was implemented in this algorithm due to its simplicity and effectiveness. In this cooling schedule, the temperature progression is based on the law of *geometric decay*, described by the function

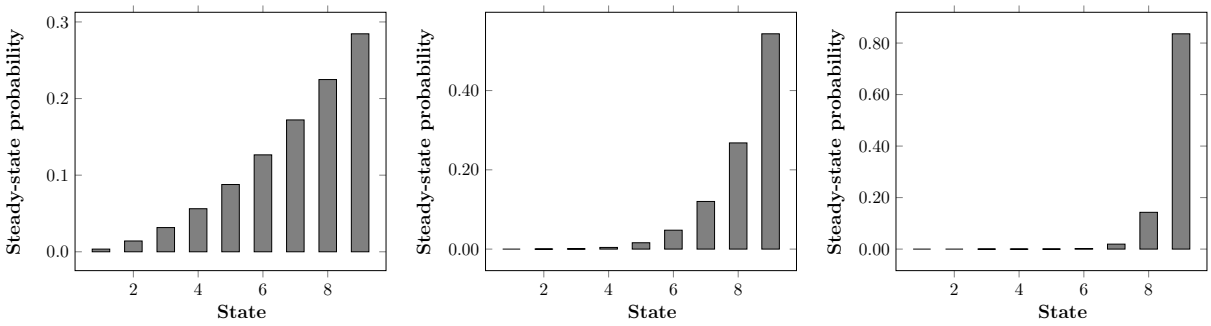
$$T_{c+1} = \pi T_c,$$

[9]Another successful class of cooling schedules are *adaptive* schedules, in which regular feedback is received from the algorithm as a means to evaluate what the next decrement in temperature should be [152]. Such schedules, however, require a certain parameter determining the change in objective function value at the end of any given epoch to be defined *a priori*. Although such a change is easily measurable when solving single-objective optimisation problems, it is not obvious how this is applicable to multiobjective optimisation problems.

where T_c is the temperature of the algorithm during search epoch $c \in \{1, \ldots, C\}$ and $\pi \in (0, 1)$ is a constant called the *cooling parameter*, and where the number of epochs is an indication of the number of times that the temperature is decreased throughout the algorithm execution (*i.e.* the length of each epoch corresponds to the expected number of iterations of the Metropolis algorithm performed in the inner loop of the method of simulated annealing). This particular cooling schedule thus provides smaller decrements in temperature as the system approaches a solidified state. According to Eglese [51], typical values of π employed in practice vary between 0.8 and 0.99.

Given a fixed computation time budget, it is of course noted that thermodynamic equilibrium is more likely to be reached during any given epoch when C is set lower. Furthermore, because the stopping criterion employed in these experiments is time elapsed, an interesting property noted during the implementation of this cooling schedule is that the expected number of iterations performed during each epoch of the algorithm is determined analogously to the (fixed) fraction of the total time pre-assigned to each temperature throughout the algorithm. Finally, it must be pointed out that this cooling schedule is also functional for any computation time budget budgets, and hence does not necessarily have to be adapted as a function of the allocated computation time budget.

Because of the large number of parameters that must be defined in this algorithm, the initial temperature for all experiments was fixed at 30 (this particular initial temperature was found to work generally well) and the cooling parameter π was selected as either 0.85 or 0.95. The number of epochs C was assigned the values of 100, 400 and 700, and all four combinations of perturbation parameters mentioned above were employed, along with three different combination values for the inter-state parameters (α, β), set at $(1.5, 2)$, $(6, 6)$ and $(15, 13)$. In order to better visualise the expected proportion of the search spent in each state subspace under these particular numerical combinations, histograms corresponding to these combinations are provided in Figure 8.6. Note that, since there are two VOIs belonging to VOI inclusion sets in this scenario, the number of states in a reduced string will range from 1 to 9.



(a) $(\alpha, \beta) = (1.5, 2)$ (b) $(\alpha, \beta) = (6, 6)$ (c) $(\alpha, \beta) = (15, 13)$

FIGURE 8.6: *Illustrations of steady-state probability distributions for different parametric configurations of the exponential growth distribution approach for inter-state transformations for the lower-complexity hypothetical scenario.*

Finally, as mentioned previously, each experiment was conducted subject to both 2- and 5-minute runs. In total, 144 experiments consisting of five runs each were therefore conducted using the method of simulated annealing. The results of these experiments are tabulated in Tables 8.10–8.13.

In cases of 2-minute runs, it appears that the majority of the best results (recalling that good results for a particular experiment are obtained when high hypervolume average with moder-

		$\alpha = 1.5, \beta = 2$			$\alpha = 6, \beta = 6$			$\alpha = 15, \beta = 13$		
		$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.4, 0.1, 0.3, 0.2)$	$C = 100$	627	126 541	37 389	484	98 262	37 762	491	153 371	41 951
	$C = 400$	761	178 936	48 267	643	152 956	47 726	671	288 280	36 818
	$C = 700$	759	141 865	45 804	705	223 224	14 646	621	211 570	79 156
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.05, 0.3, 0.3, 0.35)$	$C = 100$	659	117 667	37 352	640	148 438	37 494	530	125 186	24 148
	$C = 400$	828	184 876	26 383	761	218 980	61 323	695	183 212	93 268
	$C = 700$	826	188 960	21 545	708	188 734	24 698	748	303 338	69 661
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.3, 0.1, 0.1, 0.5)$	$C = 100$	650	96 306	21 222	513	175 211	114 723	471	99 852	32 273
	$C = 400$	800	175 744	42 305	675	164 306	47 627	581	160 056	32 071
	$C = 700$	833	154 278	17 360	709	189 864	29 504	614	187 482	49 202
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.1, 0.05, 0.1, 0.75)$	$C = 100$	671	139 628	35 107	538	121 324	31 739	505	147 770	30 270
	$C = 400$	802	204 802	76 461	691	191 796	26 956	597	194 582	55 574
	$C = 700$	868	204 108	44 952	733	267 688	42 493	668	280 062	41 038

TABLE 8.10: *Simulated annealing results for 2-minute runs and $\pi = 0.85$ for the lower-complexity hypothetical scenario.*

		$\alpha = 1.5, \beta = 2$			$\alpha = 6, \beta = 6$			$\alpha = 15, \beta = 13$		
		$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.4, 0.1, 0.3, 0.2)$	$C = 100$	317	19 539	3 330	241	16 014	2 953	178	8 832	2 798
	$C = 400$	662	127 098	28 432	505	112 076	40 928	554	190 298	44 390
	$C = 700$	699	148 048	49 934	593	171 616	57 431	588	198 946	35 043
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.05, 0.3, 0.3, 0.35)$	$C = 100$	334	26 172	12 599	219	13 525	6 123	162	11 046	1 675
	$C = 400$	725	169 530	40 720	638	178 632	50 732	570	200 916	67 730
	$C = 700$	835	226 846	52 051	652	183 720	63 185	665	253 238	33 687
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.3, 0.1, 0.1, 0.5)$	$C = 100$	326	23 137	7 147	188	13 845	5 513	166	8 114	2 664
	$C = 400$	726	137 226	24 550	597	153 513	45 143	462	137 480	29 957
	$C = 700$	753	140 878	16 689	608	160 734	63 905	598	186 242	20 950
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.1, 0.05, 0.1, 0.75)$	$C = 100$	329	20 691	3 880	205	12 824	5 274	169	10 776	2 618
	$C = 400$	703	145 116	32 588	563	137 181	61 490	523	150 272	35 068
	$C = 700$	789	171 470	42 057	655	184 902	24 834	602	178 578	19 582

TABLE 8.11: *Simulated annealing results for 2-minute runs and $\pi = 0.95$ for the lower-complexity hypothetical scenario.*

		$\alpha = 1.5, \beta = 2$			$\alpha = 6, \beta = 6$			$\alpha = 15, \beta = 13$		
		$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.4, 0.1, 0.3, 0.2)$	$C = 100$	956	286 344	53 618	796	253 370	87 571	773	419 206	88 245
	$C = 400$	1 100	434 460	57 291	849	306 142	92 256	870	485 516	94 890
	$C = 700$	1 217	404 480	62 418	921	339 234	83 066	977	591 710	101 824
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.05, 0.3, 0.3, 0.35)$	$C = 100$	998	303 302	55 437	889	318 160	89 570	863	348 088	59 600
	$C = 400$	1 259	427 628	108 861	867	288 910	142 763	1 003	504 674	180 180
	$C = 700$	1 217	460 870	157 961	1 044	382 206	38 402	1 074	534 788	85 664
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.3, 0.1, 0.1, 0.5)$	$C = 100$	984	271 256	48 883	833	253 646	31 323	750	270 120	58 141
	$C = 400$	1 215	344 490	83 010	1 053	440 212	63 380	873	339 180	83 537
	$C = 700$	1 221	366 180	26 325	978	356 434	55 281	925	435 730	55 918
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.1, 0.05, 0.1, 0.75)$	$C = 100$	1 039	322 976	29 312	817	313 310	39 766	780	357 480	59 122
	$C = 400$	1 229	528 972	173 492	968	461 708	244 653	970	544 514	118 064
	$C = 700$	1 360	615 814	114 028	1 051	555 888	240 981	918	588 020	139 655

TABLE 8.12: *Simulated annealing results for 5-minute runs and $\pi = 0.85$ for the lower-complexity hypothetical scenario.*

		$\alpha = 1.5, \beta = 2$			$\alpha = 6, \beta = 6$			$\alpha = 15, \beta = 13$		
		$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.4, 0.1, 0.3, 0.2)$	$C = 100$	459	43 358	3 548	281	28 341	13 050	239	22 983	9 355
	$C = 400$	1 019	344 238	66 285	725	230 416	132 141	835	365 148	113 186
	$C = 700$	1 101	431 830	104 796	908	440 758	95 330	904	480 896	61 243
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.05, 0.3, 0.3, 0.35)$	$C = 100$	486	50 045	14 936	299	31 392	10 539	241	26 941	3 979
	$C = 400$	1 091	342 456	47 280	921	404 880	99 123	900	479 788	88 299
	$C = 700$	1 249	538 682	61 074	994	474 582	122 235	918	505 292	202 495
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.3, 0.1, 0.1, 0.5)$	$C = 100$	470	46 901	12 897	282	29 408	7 076	218	15 094	5 058
	$C = 400$	1 102	378 806	55 200	833	216 343	112 072	767	358 108	85 350
	$C = 700$	1 215	463 120	89 088	899	350 948	108 352	902	502 988	66 233
$(p_{wr}, p_{brs}, p_{brd}, p_{is}) = (0.1, 0.05, 0.1, 0.75)$	$C = 100$	487	50 818	12 561	299	32 066	8 166	221	19 953	7 414
	$C = 400$	1 164	453 108	49 013	925	369 488	48 308	804	358 998	71 627
	$C = 700$	1 308	503 632	83 754	886	394 294	170 526	879	494 180	112 299

TABLE 8.13: *Simulated annealing results for 5-minute runs and $\pi = 0.95$ for the lower-complexity hypothetical scenario.*

ate standard deviations are achieved) are found in cases where $\pi = 0.85$, suggesting that the temperature is not decreased fast enough whenever π is set to 0.95 (with a few exceptions). As expected, this handicap is more pronounced whenever C is set lower, as the rate at which the temperature decreases during execution of the algorithm depends on both the cooling parameter and the number of epochs. Moreover, it was observed in the majority of experiments that increasing the number of epochs produces better results for both cooling parameters. In addition, the majority of the best results are also found in experiments for which $(\alpha, \beta) = (15, 13)$ suggesting that, given a relatively limited run time of 2 minutes, it seems to be more beneficial to direct the search process towards investigating higher-state solutions.

In cases of 5-minute runs, on the other hand, the majority of good results is more closely distributed between the two cooling parameters. This observation suggests that, even if the temperature is not decreased fast enough in certain experiments, a significantly larger amount of time is nevertheless still spent in the system at low temperatures (in contrast to the 2-minute runs). To be more specific, the duration of each epoch in the 5-minute runs is approximately 2.5 times longer than that those in the 2-minute runs, which also comprises epochs spent in the vicinity of intensified system solidification. In addition, it is observed that all three combinations of (α, β)-values produce some high-quality results, suggesting that algorithmic performances are not as sensitive to the nature of these parameters during 5-minute runs in contrast to 2-minute runs.

Finally, it appears in all cases that a (relatively) large archive is a necessary requirement for generating high-quality non-dominated fronts. The converse is, however, not always true, as it is observed that certain experiments produce large archives but perform relatively poorly.

8.4.3 NSGA-II results

In the multiobjective genetic algorithm, an offspring solution is mutated according to a probability specified by the mutation rate p_m. The mutation process is performed analogously to the intra-route swap transformation employed in the multiobjective simulated annealing approach of the previous section.

The parameter O_{max}, associated with the maximum number of attempts that may be performed when creating feasible offspring chromosomes during the crossover procedure (as described in §7.6.3), is fixed to five, and the marker function, responsible for setting the initial crossover delimiters (cuts) at specific locations within the parent solution strings, is configured as

$$f_M(L_1, L_2) = \begin{cases} \text{rand}(\min\{L_1, L_2\} - 1, \min\{L_1, L_2\}), & L_1, L_2 > 1, \quad L_1 \neq L_2, \\ \text{rand}(\frac{L_1}{2}, \frac{3L_1}{4}), & L_1, L2 > 1, \quad L_1 = L_2, \\ 0, & \text{otherwise}, \end{cases}$$

where the function $\text{rand}(a, b)$ denotes a random integer sampled from a uniform distribution on the interval of positive integer values (a, b) with $a \leq b$, while $L_1 = |\boldsymbol{p}_1|$ and $L_2 = |\boldsymbol{p}_2|$ refer to the reduced string sizes of the selected parent solutions.

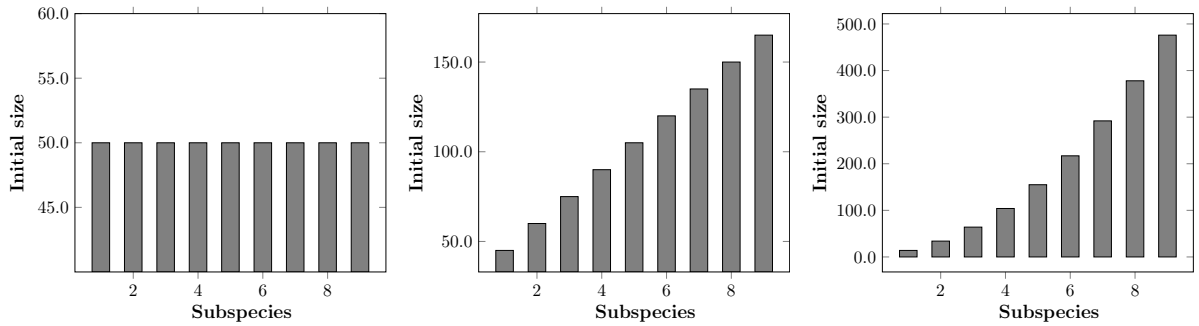
As discussed in §7.6.2, the initial population is constructed by allocating a specific number of individuals to each subspecies (*i.e.* individuals of certain string sizes). As the number of distinct string sizes may be relatively large, and the distribution of individuals within initial subspecies may vary from one MLE response selection problem instance to another, however, it may be too tedious and time-consuming for the decision maker to assign such numbers to every subspecies. Hence, the function

$$G_s = \lambda + \sigma s^{\mu}, \qquad s \in \{1, \ldots, \tilde{n}_\tau\}$$

is adopted as a means of populating the anterior stage of the evolution process, where G_s represents the initial generation size of subspecies s. The (fixed) population size is therefore computed as

$$P_{size} = 2\sum_{s=1}^{\tilde{n}_\tau} G_s,$$

where $\tilde{n}_\tau$ denotes the number of VOIs in a reduced solution string during time stage τ (analogously to the number of states in simulated annealing). According to this initial population configuration, low values of σ and μ therefore allow for a more evenly spread of solutions amongst the population subspecies. Furthermore, the size of the population, together with the rate of change from the size of one subspecies to the next, is extremely sensitive to the parameter μ, particularly in cases where $\tilde{n}$ is large. Examples of initial population configurations with respect to each subspecies for 5-minute runs are illustrated in Figure 8.7. Here, Figure 8.7(a) represents a uniformly distributed initial population amongst the subspecies, while Figure 8.7(b) represents one that is linearly distributed and Figure 8.7(c) one that is exponentially distributed.



(a) $(\lambda, \sigma, \mu) = (50, 0, 1)$ (b) $(\lambda, \sigma, \mu) = (15, 15, 1)$ (c) $(\lambda, \sigma, \mu) = (0, 3, 2.2)$

FIGURE 8.7: *Illustrations of initial population subspecies distribution of the NSGA-II for 5-minute runs for the lower-complexity hypothetical scenario.*

Two interesting differentiations with respect to subspaces exploration and non-dominated front generation between the method of simulated annealing and NSGA-II should be pointed out. First, contrary to the method of simulated annealing, in which the the search repartition is controlled by inter-state transformations throughout the algorithm, the NSGA-II is only able to control the state distribution of its initial population, after which individuals are evolved to the next generations on the basis of merit (fitness) only, and not based on the state of their underlying string constitution. Secondly, unlike the method of simulated annealing, where the archive can grow "indefinitely" as the search progresses, the size of the final non-dominated front generated by the NSGA-II is expected to be significantly handicapped by the size of its initial population.

As observed in the case of the experiments performed in the context of the method of simulated annealing in the previous section, non-dominated fronts are not necessarily bounded in quality (according to the hypervolume measure) by the number of solutions that they contain. Because there exists a very strong negative correlation between small non-dominated fronts and hypervolume values, however, it is nevertheless acknowledged that the population size ought to be set large enough for specific run times. Different sets of initial population parameters consequently have to be selected for each computational budget in order to improve the chances of identifying high-quality combinations of parameters for each computation time budget configuration. As with the method of simulated annealing, each experiment involving the NSGA-II was con-

ducted under both 2- and 5-minute computational budgets. The results of these experiments are tabulated in Tables 8.14 and 8.15.

As predicted earlier, it is apparent that the performance of the search suffers significantly if the generation size is set too low. This may be observed, for example, whenever the combination of parameters $(\lambda, \sigma, \mu) = (30, 0, 1)$ is employed for 2-minute runs, and whenever the combination of parameters $(\lambda, \sigma, \mu) = (50, 0, 1.5)$ is employed for 5-minute runs.

Unfortunately, no statistical evidence is observed from these results suggesting that the mutation rate has a significant impact on algorithmic performances. This is believed to be caused by the exploratory nature of the crossover operator employed, as described in §7.6.3. Moreover, the impact of the mutation operator and the mutation rate on the performance indicator is anticipated to become more insignificant as the population size increases, as a relatively large initial population is more likely to incorporate solutions spanning all or most feasible subsets of the decision space (which may render the diversification benefits associated with this operator useless).

In general, it was found that the NSGA-II exhibits very good convergence properties (that is, the level of improvement from one non-dominated front to the next becomes very small towards the end of a search run) so long as the number of generations performed is large enough, especially when a large population size is selected. Moreover, it was observed that convergence typically takes place between 45 and 75 generations[10]. The initial population size should therefore not be set too large either, as the number of generations performed by the algorithm may then be too small for the first non-dominated front to reach a state of convergence or near-convergence. Furthermore, the sizes of the non-dominated fronts generated in most experiments were found to be very close to one another, indicating a certain level of consistency in the functionality of the algorithm over various combinations of parameter values.

Finally, it was stressed in §7.6.2 that there exists a risk of experiencing subspecies extinctions during the course of the search process. Fortunately, it was observed throughout these experiments that the NSGA-II was able to prevent such undesirable outcomes in most cases. The reason for this, it was found, is simply that non-dominated solutions in this MLE response selection problem instance are well distributed amongst the species subspaces. In other words, each species subspace contains a relatively large number of solutions (with respect to the size of the subspace) that are not dominated by any solutions in any other state subspaces.

8.4.4 Hybrid metaheuristic results

As described in §7.7, the proposed hybrid metaheuristic is, in most aspects, structurally identical to the NSGA-II. The only difference evolves around the unconventional mutation process employed. Here, mutated solutions are accepted according to the Metropolis rule of acceptance of the archived simulated annealing algorithm. As in the method of simulated annealing, this rule of acceptance becomes stricter as the number of generations performed increases. The aim in testing this hybrid metaheuristic experimentally is ultimately to determine whether a solution-exploitation sub-loop governed by this rule of acceptance is capable of producing better results than the NSGA-II in which standard mutation procedures are adopted (note that this approach differs from employing a hypermutation operator, where the mutation rate is altered as a function of the current generation of the underlying genetic algorithm).

Due to the rarity of employing hybrid search methods of this kind in the literature as well as

[10]Convergence evaluation was estimated by recording the average change along each objective function axis amongst the non-dominated front individuals from one generation to the next.

		$p_m = 0.05$			$p_m = 0.15$			$p_m = 0.3$		
	P_{size}	$\overline{\mathcal{F}}_1\|$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\overline{\mathcal{F}}_1\|$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\overline{\mathcal{F}}_1\|$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$\lambda = 0$, $\sigma = 2$, $\mu = 2.2$	2 318	1 265	237 922	22 330	1 283	252 618	32 138	1 283	261 120	34 417
$\lambda = 0$, $\sigma = 25$, $\mu = 1$	2 702	1 524	247 192	32 638	1 524	262 238	118 721	1 529	282 650	34 076
$\lambda = 5$, $\sigma = 10$, $\mu = 1.5$	2 932	1 683	252 494	33 090	1 560	232 678	65 055	1 653	275 014	49 575
$\lambda = 10$, $\sigma = 10$, $\mu = 1$	1 262	691	130 744	16 538	699	147 558	39 277	681	140 104	18 202
$\lambda = 10$, $\sigma = 25$, $\mu = 1$	2 882	1 630	280 348	45 226	1 656	275 508	43 100	1 660	267 486	41 076
$\lambda = 30$, $\sigma = 0$, $\mu = 1$	542	306	21 058	1 775	306	25 698	4 370	303	27 062	4 704
$\lambda = 70$, $\sigma = 0$, $\mu = 1$	1 262	680	147 682	25 378	696	125 488	23 385	685	129 561	26 035
$\lambda = 110$, $\sigma = 0$, $\mu = 1$	1 982	1 072	276 794	53 516	1 070	265 014	68 266	1 042	256 020	25 558

TABLE 8.14: *NSGA-II results for 2-minute runs for the lower-complexity hypothetical scenario.*

	P_{size}	$p_m = 0.05$			$p_m = 0.15$			$p_m = 0.3$		
		$\overline{\mathcal{F}_1}$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\overline{\mathcal{F}_1}$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\overline{\mathcal{F}_1}$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$\lambda = 0, \sigma = 3, \mu = 2.2$	3 470	1 889	641 474	140 767	1 883	638 490	85 824	1 874	615 676	164 672
$\lambda = 0, \sigma = 35, \mu = 1$	3 782	2 072	778 168	271 092	2 083	674 274	121 000	2 048	763 354	155 385
$\lambda = 10, \sigma = 15, \mu = 1.5$	4 438	2 508	632 336	85 496	2 512	708 834	144 128	2 505	709 062	81 439
$\lambda = 15, \sigma = 15, \mu = 1$	1 892	1 028	288 776	50 705	1 035	276 424	41 734	1 019	305 564	27 670
$\lambda = 15, \sigma = 30, \mu = 1$	3 512	1 898	677 542	143 719	1 916	705 052	99 131	1 891	719 816	122 324
$\lambda = 50, \sigma = 0, \mu = 1$	902	495	77 130	14 281	497	85 197	20 516	500	80 275	9 220
$\lambda = 100, \sigma = 0, \mu = 1$	1 802	984	262 854	29 009	979	224 854	31 403	970	249 240	54 667
$\lambda = 150, \sigma = 0, \mu = 1$	2 702	1 460	495 766	50 981	1 458	508 034	54 423	1 459	496 750	72 001

TABLE 8.15: *NSGA-II results for 5-minute runs for the lower-complexity hypothetical scenario.*

to the large quantity of required user-defined parameters, the experiments performed in this section had to be carefully considered in respect of selecting a small number of combinations that may work most efficiently. In the outer-loop of the experiments performed using this hybrid metaheuristic, four different configurations of initial populations and three configurations of mutation rates are employed. In the inner-loop of this hybrid, two inter-route transformations are performed on mutation candidate solutions with equal probabilities, while two different cooling schedules are tested (namely a fast one and a slow one). All parameters in this section are defined as in §8.4.2 and §8.4.3. Additionally, due to the poor convergence results of NSGA-II under 2-minute runs (see §8.4.2), these experiments were only tested in the context of 5-minute runs. The results of these experiments are tabulated in Tables 8.16 and 8.17.

As hinted at earlier for the NSGA-II, a drawback of using population-based algorithms in large discrete optimisation problems with many objectives is that the population size will, on the one hand, need to be large enough in order to access a large proportion of a large quantity of Pareto or near-Pareto-optimal solutions while, on the other hand, not be too large so as to allow the algorithm to converge within a limited computation time budget. The hybrid experiments were tested with population sizes (P_{size} values) of the order of 1 000 individuals.

It was observed that experiments containing 4 718 individuals struggle to achieve convergence and, in certain cases, deteriorate in quality in comparison to experiments that employ fewer individuals, but that allow evolution to take place over a longer period of time, even though the size of their final non-dominated fronts are clearly smaller. On the other hand, it appears that the algorithmic performances deteriorate whenever population sizes of 2 710 are intialised, even though these possess better convergence properties. This is believed to be mainly caused by the restriction of initial population sizes imposed on the size of the final non-dominated front.

The inner-loop mutation process is expected to consume a significant amount of computation time budget in comparison to standard mutation procedures, and this is, of course, more pronounced whenever a high mutation rate (p_m) is employed. This is particularly apparent when considering any three experiments from the same population constitution and cooling schedule, where the average number of generations completed by the algorithm ($\overline{N}_{Gen}$ values) can vary as much as 30% from one mutation setting to the next. Interestingly, however, this does cause the algorithm to perform poorly. Indeed, as a result of the benefits brought along by the controlled acceptance of improving solutions, certain parametric configurations benefit more from using this adapted inner-loop more frequently throughout the execution of the algorithm (*i.e.* higher mutation) even though they may not be allocated sufficient time to converge.

Lastly, there is enough statistical evidence to suggest that, given any population size and mutation rate configuration, a slow cooling schedule produces more generations. This is because the mutation process is allowed more than one attempt (see the parameter I_{max} in Algorithm 7.7) to find a feasible improving solution during the process of exploiting the current solution. In these experiments, this parameter was set to a value of three. And so, because a fast cooling schedule will spend a greater proportion of the search conducting moves in "strict" acceptance zones, more attempts will, on average, be required to generate improving solutions. This, in turn, means that a greater amount of time will be spent in the inner-loop of the algorithm, and so fewer generations may be completed during the course of the search. Furthermore, it is believed that the algorithm is expected to perform better in limited computation time budget whenever it is encouraged to explore more (which is what a slow cooling schedule entails in the method of simulated annealing).

		$p_m = 0.05$				$p_m = 0.15$				$p_m = 0.3$			
	P_{size}	$\overline{N}_{Gen}$	$\overline{\mathcal{F}}_1\|$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\overline{N}_{Gen}$	$\overline{\mathcal{F}}_1\|$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\overline{N}_{Gen}$	$\overline{\mathcal{F}}_1\|$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$\lambda = 5$, $\sigma = 15$, $\mu = 1$	1 712	102	909	224 920	23 146	83.4	901	250 884	9 346	906	67.8	270 168	43 401
$\lambda = 5$, $\sigma = 20$, $\mu = 1.1$	2 710	65.8	1 427	443 812	73 899	56.2	1 415	478 124	83 333	47.4	1 445	430 352	91 064
$\lambda = 5$, $\sigma = 19$, $\mu = 1.295$	3 708	48.2	2 002	577 510	88 354	41.8	2 010	551 550	103 712	34.6	2 078	448 590	90 108
$\lambda = 5$, $\sigma = 17.45$, $\mu = 1.465$	4 718	41	2 641	645 038	153 990	37.2	2 671	625 298	189 862	31.4	2 176	416 756	49 217

TABLE 8.16: *Hybrid metaheuristic results for a fast cooling schedule, with $C = 700$, $T_0 = 30$ and $\pi = 0.85$ for the lower-complexity hypothetical scenario.*

		$p_m = 0.05$				$p_m = 0.15$				$p_m = 0.3$			
	P_{size}	$\overline{N}_{Gen}$	$\overline{\mathcal{F}}_1\|$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\overline{N}_{Gen}$	$\overline{\mathcal{F}}_1\|$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$	$\overline{N}_{Gen}$	$\overline{\mathcal{F}}_1\|$	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$\lambda = 5$, $\sigma = 15$, $\mu = 1$	1 712	108.2	912	261 432	57 889	87	903	261 036	48 873	69.4	903	310 264	90 404
$\lambda = 5$, $\sigma = 20$, $\mu = 1.1$	2 710	70.4	1 422	523 614	77 391	59.8	1 420	431 904	70 912	49.6	1 428	445 642	86 637
$\lambda = 5$, $\sigma = 19$, $\mu = 1.295$	3 708	52.2	1 985	579 116	54 335	44.4	2 021	680 736	76 929	38	2 063	564 084	112 401
$\lambda = 5$, $\sigma = 17.45$, $\mu = 1.465$	4 718	40	2 632	529 418	68 828	36	2 551	709 694	253 940	31.6	2 142	548 942	84 965

TABLE 8.17: *Hybrid metaheuristic results for a slow cooling schedule, with $C = 300$, $T_0 = 30$ and $\pi = 0.85$ for the lower-complexity hypothetical scenario.*

8.4.5 Augmented multistart simulated annealing results

The aim in this section is to investigate whether parallel processing computing attempts are able to make a significant difference in terms of finding high-quality non-dominated sets of solutions in comparison to their sequential processing counterparts. To this effect, the augmented multistart archived simulated annealing algorithm of §7.8 was implemented within a multistart paradigm, although it is acknowledged that a multistart genetic algorithm may also work very well. The processors are configured with combinations of the simulated annealing parametric configurations (deemed to perform relatively well) of §8.4.2.

In addition to the various combinations of archived simulated annealing parameters that may be used on each processor, a new parameter also has to be considered in these experiments, namely the number of processors operating in parallel. This is a particularly important decision in the practical implementation of any real-life DSS, where a trade-off may be pursued between the expected increased performance linked to the addition of a processor and the additional costs of setting up, operating and maintaining an additional processor. Aside from the obvious claim that multistart computing will always perform at least as well as its best-performing processor, a system engineer will ask himself an important question in this regard, namely what expected marginal increase in the quantity and quality of non-dominated solutions may be expected when an additional parallel processor is to be employed in addition to the current ones. This is, however, not trivial question to answer, as parallel computing involving a fixed number of processors may perform significantly differently given varying complexity scenarios and varying algorithm run times.

In order to simulate the parallel computing efficiency in respect of the query above, a basic *multiobjective parallel computing efficiency measure*, defined as

$$\vartheta = \frac{\overline{H}_{vol}}{\overline{H}^{\mathcal{P}}_{vol}} \left(\frac{|\mathcal{A}|}{\sum_{i=1}^{N} |\mathcal{A}_i|} \right) = \vartheta(\overline{H}_{vol}) \vartheta(|\mathcal{A}|) \in (0, 1],$$

is adopted for the multistart experiments conducted in this chapter. Here, N is the number of processors employed, $\overline{H}^{\mathcal{P}}_{vol}$ is an approximation of the hypervolume quantity of the true Pareto front, while $\mathcal{A}_i$ denotes the archive computed by Processor i and $\mathcal{A}$ denotes the augmented archive.

In the above formulation, the metric $\vartheta(|\mathcal{A}|)$, defined on the interval $\left[\frac{1}{N}, 1\right]$, is a quantification of the effectiveness of the combined effort of parallel processors to uncover a large number of non-dominated solutions when pooled together, after duplicate and dominated solutions have been removed from this pool. In other words, adopting additional processors is more justified if there are not many solutions in the intersection sets of any two parallel processor archives. Moreover, the metric $\vartheta(\overline{H}_{vol})$, defined on $(0, 1]$, is an approximation of the achievement percentage of hypervolume of the pooled archive with respect to the estimated maximum hypervolume. As mentioned before, the true Pareto front is typically not known with certainty, and so neither is its hypervolume. Noting that underestimating or overestimating its value will not affect the utility of the metric ϑ significantly in respect of comparing the performance variations when using different numbers of processors, it is assumed that $\overline{H}^{\mathcal{P}}_{vol} \approx 2\ 300\ 000$ (based on the results obtained earlier in this chapter).

Due to the limitations in computer resources and computer processing power at hand, three to six runs per experiment were conducted under a computational budget of two minutes only, the results of which are tabulated in Table 8.18. In this scenario, only two, three and four processors were tested. Moreover, all processors are assumed to be identical in computing power

and memory capacity and all parameters were defined as in §8.4.2. As expected, this multistart simulated annealing algorithm strives to attain a higher proportion of the Pareto optimal front in comparison to its unistart counterpart, attaining as much as an estimated 67% of the maximum hypervolume with just four processors.

8.4.6 Discussion

Selecting high-quality combinations of search parameter values requires a satisfactory trade-off between the average hypervolume and hypervolume standard deviation to be identified by the decision maker. Here, an experiment is said to dominate another one if both its average hypervolume and its standard deviation are larger and lower, respectively, than those of the other experiment. A risk-averse decision maker would, for example, be more interested in selecting a search process involving a combination of parameter values that is expected to produce non-dominated fronts with reasonably high hypervolumes every time and with relatively small standard deviations, so as to minimise the risk of generating a poor-quality, non-dominated front during any given run of the search process.

Using this notion as a filtering mechanism to draw out efficient parametric configurations from the multitude of experimental results obtained, a summary of recommended parameter value combinations is provided in Tables 8.19–8.23. These results may be used when solving problem instances of this type by means of the proposed unistart optimisation techniques.

It is advisable, however, that the credibility of the statistics gathered from these experiments be strengthened by performing additional runs. Additionally, relevant statistics on the performances of these optimisers should be computed, or accurate probability distributions of the hypervolume values associated with preferred combinations of parameter values should be approximated. In particular, it is inappropriate to assume that the probability distribution of the hypervolume measure may be mapped using the same type of probability distribution for all parameter combinations.

With respect to 2-minute runs, there is not enough statistical evidence to suggest that any one of the two standard search techniques performs better than the other (under any parametric configurations). It appears, however, that the NSGA-II is able to outperform the method of simulated annealing significantly when the termination criterion is set to a higher run time, provided that its initial population size is large enough.

As deduced from Figures 8.6 and 8.7, it is believed that one of the primary reasons that NSGA-II most often outperforms the method of simulated annealing within larger computational time budgets is because it better exploits solutions located in lower state subspaces as a result of the nature of its initial population subspecies configuration. It is analogously anticipated that a parallel multistart NSGA-II may strongly outperform the augmented multistart simulated annealing method within both smaller and larger computation time budgets as a result of its "fairer" allocation of the computational budget to finding non-dominated solutions located in isolated feasible solution subspaces, combined with the law of large numbers emanating from the use of numerous processors.

Notice that the results in Tables 8.19 and 8.21 for the method of simulated annealing do not contain repeated combinations of parameter values. In addition, as observed from the preliminary experiments, combinations of parameter values resulting in good results within one minute of computation time do not necessarily perform well within two minutes of computation time either (and *vice versa*). These observations therefore suggest that different combinations of parameter values will have to be selected when executing a search process within different

Processors parametric configurations					Processors sample statistics		Augmented sample statistics		Performance metrics		
N	$(p_{wr}, p_{brs}, p_{brd}, p_{is})$	(α, β)	C	π	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$\vartheta(\lvert\bar{\mathcal{A}}\rvert)$	$\vartheta(\bar{\overline{H}}_{vol})$	$\bar{\vartheta}$
2	(**2**×) $(0.4, 0.1, 0.3, 0.2)$	$(6, 6)$	700	0.95	719	232 311	1 227	544 514	0.853	0.237	0.202
	(**2**×) $(0.05, 0.3, 0.3, 0.35)$	$(15, 13)$	700	0.95	689	314 458	1 131	712 322	0.821	0.310	0.254
	(**2**×) $(0.4, 0.1, 0.3, 0.2)$	$(15, 13)$	400	0.85	642	248 748	1 106	609 688	0.861	0.265	0.228
	(**2**×) $(0.3, 0.1, 0.1, 0.5)$	$(15, 13)$	400	0.85	649	248 797	1 095	649 216	0.843	0.282	0.238
3	(**3**×) $(0.4, 0.1, 0.3, 0.2)$	$(6, 6)$	700	0.85	708	225 957	1 613	1 020 926	0.778	0.444	0.345
	(**3**×) $(0.05, 0.3, 0.3, 0.35)$	$(15, 13)$	700	0.95	691	294 927	1 583	1 208 333	0.763	0.525	0.401
	(**3**×) $(0.4, 0.1, 0.3, 0.2)$	$(15, 13)$	400	0.85	639	258 020	1 490	1 078 863	0.777	0.469	0.364
	(**3**×) $(0.3, 0.1, 0.1, 0.5)$	$(15, 13)$	700	0.85	673	265 279	1 514	1 177 867	0.750	0.512	0.384
4	(**1**×) $(0.4, 0.1, 0.3, 0.2)$	$(6, 6)$	700	0.85	651	248 193	1 838	1 430 900	0.706	0.622	0.439
	(**1**×) $(0.05, 0.3, 0.3, 0.35)$	$(15, 13)$	700	0.95							
	(**1**×) $(0.4, 0.1, 0.3, 0.2)$	$(15, 13)$	400	0.85							
	(**1**×) $(0.3, 0.1, 0.1, 0.5)$	$(15, 13)$	700	0.85							
	(**2**×) $(0.4, 0.1, 0.3, 0.2)$	$(6, 6)$	700	0.85	659	265 570	1 946	1 549 766	0.703	0.674	0.474
	(**2**×) $(0.05, 0.3, 0.3, 0.35)$	$(15, 13)$	700	0.95							

TABLE 8.18: *Augmented multistart archived simulated annealing algorithm results for the lower-complexity hypothetical scenario.*

$(p_{wr}, p_{brs}, p_{brd}, p_{is})$	(α, β)	C	π	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$(0.4, 0.1, 0.3, 0.2)$	$(6, 6)$	700	0.85	223 224	14 646
$(0.05, 0.3, 0.3, 0.35)$	$(15, 13)$	700	0.95	253 238	33 687
$(0.4, 0.1, 0.3, 0.2)$	$(15, 13)$	400	0.85	288 280	36 818
$(0.3, 0.1, 0.1, 0.5)$	$(15, 13)$	700	0.85	303 338	69 661

TABLE 8.19: *Recommended simulated annealing parametric configurations for 2-minute runs for the lower-complexity hypothetical scenario.*

(λ, σ, μ)	p_m	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$(0, 2, 2.2)$	0.05	237 922	22 330
$(110, 0, 1)$	0.30	256 020	25 558
$(0, 25, 1)$	0.30	282 650	34 076

TABLE 8.20: *Recommended NSGA-II parametric configurations for 2-minute runs for the lower-complexity hypothetical scenario.*

$(p_{wr}, p_{brs}, p_{brd}, p_{is})$	(α, β)	C	π	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$(0.3, 0.1, 0.1, 0.5)$	$(1.5, 2)$	700	0.85	366 180	26 235
$(0.05, 0.3, 0.3, 0.35)$	$(6, 6)$	700	0.85	382 206	38 402
$(0.1, 0.05, 0.1, 0.75)$	$(1.5, 2)$	400	0.95	453 108	49 013
$(0.05, 0.3, 0.3, 0.35)$	$(1.5, 2)$	700	0.95	538 682	61 074
$(0.4, 0.1, 0.3, 0.2)$	$(15, 13)$	700	0.85	591 710	101 824
$(0.1, 0.05, 0.1, 0.75)$	$(1.5, 2)$	700	0.85	615 814	114 028

TABLE 8.21: *Recommended simulated annealing parametric configurations for 5-minute runs for the lower-complexity hypothetical scenario.*

(λ, σ, μ)	p_m	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$(150, 0, 1.5)$	0.05	495 766	50 981
$(150, 0, 1.5)$	0.15	508 034	54 423
$(10, 15, 1.5)$	0.30	709 062	81 439
$(15, 30, 1)$	0.30	719 816	122 324
$(0, 35, 1)$	0.30	763 354	155 385
$(0, 35, 1)$	0.05	778 168	271 092

TABLE 8.22: *Recommended NSGA-II parametric configurations for 5-minute runs for the lower-complexity hypothetical scenario.*

(λ, σ, μ)	p_m	Cooling speed	$\overline{H}_{vol}$	$s(\overline{H}_{vol})$
$(5, 17.45, 1.465)$	0.3	fast	416 756	49 217
$(5, 19, 1.295)$	0.05	slow	579 116	54 335
$(5, 19, 1.295)$	0.15	slow	680 736	76 929
$(5, 17.45, 1.465)$	0.15	slow	709 694	253 940

TABLE 8.23: *Recommended Hybrid parametric configurations for 5-minute runs for the lower-complexity hypothetical scenario.*

computational budgets. In addition to improving the hypervolume statistic by means of additional runs and experimenting with other combinations of parameter values (preferably in the vicinity of the recommended combinations of parameter values), future work in this area may focus on analysing the performance of the method of simulated annealing subject to different computational budgets

Although not pursued in the parallel computing experiments conducted in this chapter, it is of course perfectly acceptable to strategically combine the results of processors employed to implement the method of simulated annealing to others employed in implementing the NSGA-II. This may be achieved similarly to employing the augmented non-dominated front technique of Algorithm 7.9. A visual test of the merged non-dominated fronts of two processors, each configured to operate under one of these two algorithms, is portrayed in Figure 8.8. Although this is just one of countless possible experimental combinations, it is easy to see that both fronts only rarely intersect, confirming once more that incorporating parallel computing as part of an MLE response selection DSS may be a good idea.

Finally, due to the type of stopping criterion used in the search processes of these experiments, it is acknowledged that the ability of a search process to perform a certain number of iterations/generations under a fixed time limit depends on critical factors such as the level of programming language[11] as well as the processing power of the computer(s) employed. Consequently, it is suggested that most of the experiments reported in this chapter ought to perform significantly better when higher processing power and/or lower complexity languages is implemented.

8.5 A higher-complexity hypothetical scenario

The aim in this section is to assess the performances of some of the proposed search methods in a more complex environment. More specifically, the hypothetical scenario considered in this section has been designed to test the ability of selected algorithms to uncover acceptable subsets of non-dominated solutions from a much larger domain space under a similar computation time budget.

Threat type	MLE resource i
1	(0.5, 0.5)
2	(0.7, 0.3)
3	(0.5, 0.5)
4	(0.2, 0.8)
5	(0.7, 0.3)
6	(0.7, 0.3)
7	(0.8, 0.2)
8	(0.7, 0.3)
9	(0.8, 0.2)
10	(1, 0.1)

TABLE 8.24: *MLE resource i counter-threat ability levels, as first entries, and expected service times (hours), as second entries.*

With only very slight modifications, the hypothetical scenario introduced in this section is

[11]It has previously been shown that the time required to perform a fixed series of tasks in a computational evaluation process may be reduced by more than ten times when a relatively low level of programming language is employed [16].

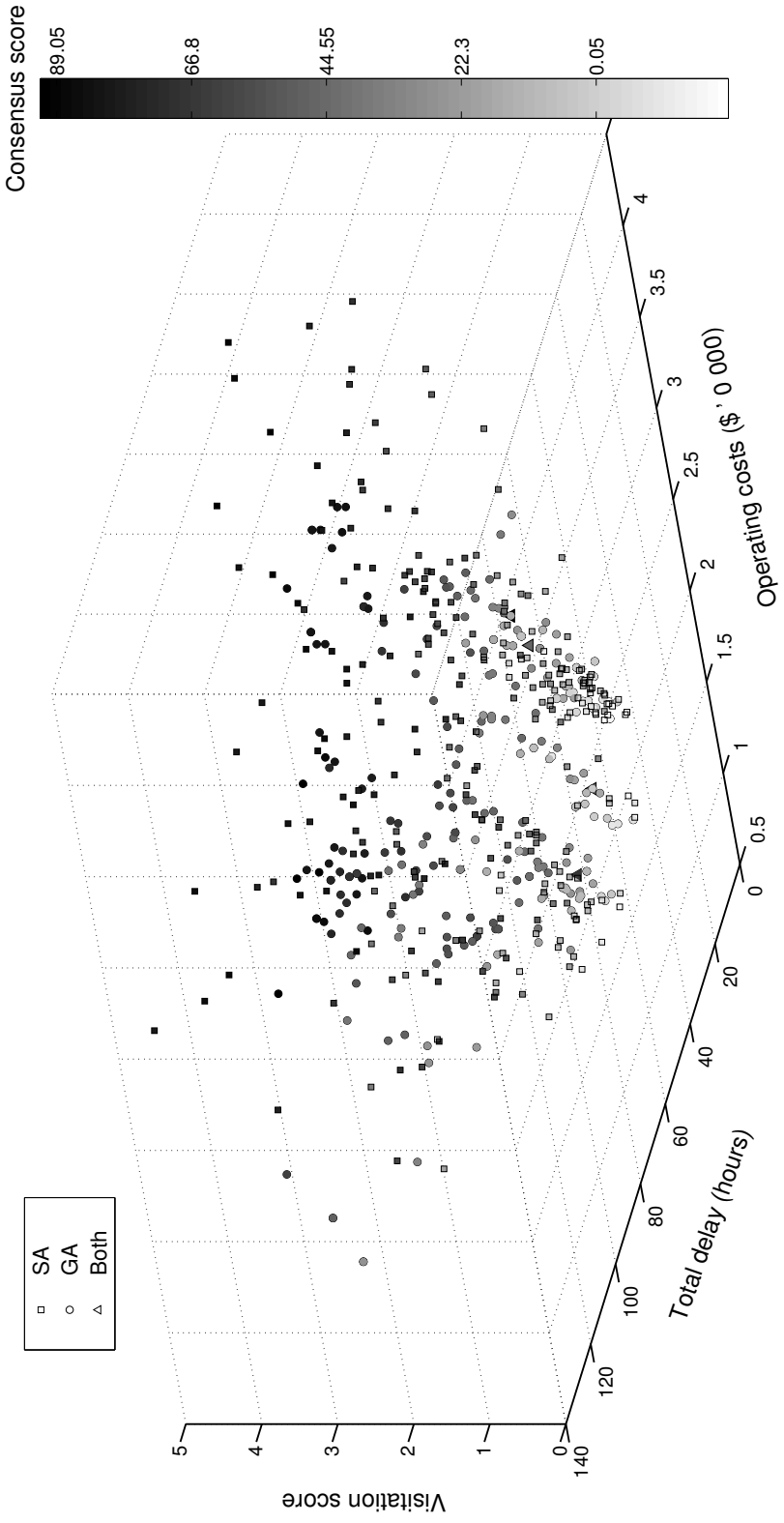


FIGURE 8.8: *Merged non-dominated fronts obtained by two processors employed to implement the NSGA-II and the method of simulated annealing, respectively, for the lower-complexity hypothetical scenario.*

essentially an extension of the lower-complexity scenario of §8.3. This extension consists in introducing an additional ten VOIs and another MLE resource using the same mathematical model as the one in §8.3.3. All modified/additional information in this hypothetical scenario are presented in Tables 8.24–8.28 in a similar display as those of §8.3. In addition, the physical elements of this hypothetical scenario at time $t = 0$ are laid out visually as shown in Figure 8.9.

Attribute	MLE resource i
Average speed (km/h)	60
Setup costs (\$)	1 000
Setup time (h)	0.4
Travelling costs $(\$\backslash km)$	30
Distance autonomy (km)	1 500
Time Autonomy (h)	100
B_1 Allocation status (β_{1i})	0
B_2 Allocation status (β_{2i})	1
B_3 Allocation status (β_{3i})	1
B_4 Allocation status (β_{4i})	1
Current location	$(255, 110)$
Distance autonomy (kms)	1 200
Time autonomy $(hours)$	95
γ_k	0
Base exclusion set	$\{B_2\}$
Patrol circuit inclusion set	$\{P_4, P_6, P_7\}$

TABLE 8.25: *MLE resource i attribute characteristics (found above the mid-section in the table) and time stage parameters (found below the mid-section in the table).*

MLE resource	a	b	c	d	e	f	g	h	i
VOI exclusion sets	$\emptyset$	$\emptyset$	$\{9\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{12\}$	$\emptyset$	$\emptyset$
VOI inclusion sets	$\{18\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{19\}$	$\{2\}$	$\emptyset$	$\emptyset$	$\{13\}$

TABLE 8.26: *VOI exclusion sets and VOI inclusion sets.*

	Patrol Circuit						
	P_1	P_2	P_3	P_4	P_5	P_6	P_7
MLE resource i	(550,60)	(400,60)	(600,50)	(450,50)	(475,50)	(275,40)	(375,50)

TABLE 8.27: *MLE resource i patrol circuit distance autonomy thresholds (km), as first entries, and patrol circuit time autonomy thresholds (hours), as second entries.*

8.5.1 Pre-optimisation analysis

Prior to launching any analytical optimisation experiments it is necessary to gauge key characteristics pertaining to the domain, search and objective spaces of the combinatorial optimisation problem under consideration. Given that this scenario involves nine MLE resources and twenty one VOIs, and adopting the formulation in §8.2.3, the domain size of this problem is approximated at a respectable 4.508×10^{26} solutions. Again, it is stressed that model management

Entity	VOI preference ordered set	Ideal quantity
1	$\mathcal{O}_1 = \{9, 13, 12, 18, 7, 19, 3, 15, 5, 21, 6, 20, 14, 8, 1, 17, 16, 10, 2, 11, 4\}$	$N_1 = 6$
2	$\mathcal{O}_2 = \{14, 21, 16, 19, 1, 3, 15, 8, 18, 10, 5, 7, 9, 12, 6, 11, 13, 4, 17, 20, 2\}$	$N_2 = 4$
3	$\mathcal{O}_3 = \{11, 4, 13, 17, 2, 19, 20, 14, 1, 16, 18, 5, 8, 6, 10, 21, 3, 9, 15, 7, 12\}$	$N_3 = 7$

TABLE 8.28: *Input data received from the decision making entities.*

VOI	Location	Estimated itinerary	Threat probabilities	Delay (hours)
12	$(-375, 150)$	$(-375, 150 - 10t)$	$(0, 0, 0, 0.1, 0, 0.3, 0.3, 0, 0, 0.3)$	0.5
13	$(250, 125)$	$(250 - 10t, 125)$	$(0.1, 0.3, 0, 0.6, 0, 0, 0, 0, 0, 0)$	1.5
14	$(-225, 135)$	$(-225 - 5t, 135 - 5t)$	$(0, 0, 0.4, 0.5, 0, 0, 0, 0, 0, 0.1)$	0.5
15	$(50, 250)$	$(50 + 15t, 250 + 10t)$	$(0, 0.3, 0, 0, 0, 0.3, 0, 0, 0.4, 0)$	0.25
16	$(-180, 310)$	$(-180, 310)$	$(0, 0, 0, 0.7, 0, 0, 0, 0, 0, 0.3)$	0.25
17	$(110, 170)$	$(110 - 5t, 170 - 10t)$	$(0.7, 0.2, 0, 0, 0, 0, 0, 0, 0, 0)$	1
18	$(-300, 125)$	$(-300 - 10t, 125 + 15t)$	$(0, 0, 0, 0, 0.1, 0, 0.4, 0.4, 0.1, 0)$	0.75
19	$(-90, 250)$	$(-90 + 5t, 250 + 20t)$	$(0, 0, 0, 0, 0.4, 0.4, 0, 0, 0, 0.2)$	1.5
20	$(200, 115)$	$(200, 115)$	$(0.15, 0.65, 0, 0, 0, 0, 0, 0, 0, 0.2)$	0.75
21	$(-25, 270)$	$(-25 + 5t, 270 - 15t)$	$(0.1, 0.1, 0.2, 0.1, 0, 0.1, 0.1, 0.1, 0.1, 0.1)$	0

TABLE 8.29: *Input data for additional VOIs.*

features, algorithm configuration and intelligent exploration moves are fortunately able to focus the search space to a very restricted subset of this rather large domain. It is nevertheless acknowledged that a much larger search space than that of the hypothetical problem of §8.3 has to be considered.

After conducting numerous test runs, numerical bounds associated with each objective function were approximated for hypervolume calculation purposes, which are shown in Table 8.30. The normalisation of objective functions and the selection of a reference point for the purpose of hypervolume evaluation were conducted similarly as in §8.4.1.

Due to the large number of VOIs in contrast to the number of available MLE resources, combined with the presence of several strict MLE resource autonomy constraints, it was also deemed necessary to investigate the feasibility status of subspaces associated with higher solution states. This was done in order to assess the potential waste associated with the repartition of computational budget whilst searching for good solutions in higher-state subspaces. Interestingly, for instance, it was discovered that there (presumably) exists no feasible solutions visiting a total of twenty or twenty one VOIs in this scenario. In addition, it was also observed that feasible solutions visiting a total of eighteen or nineteen VOIs were rarely uncovered. Finding good solutions visiting a total of seventeen VOIs satisfying the MLE response selection decision entities'

Objective	Type of bound	Numerical bound
Visitation score	upper bound	8.422
Total delay	lower bound	3.95 (hours)
Operating costs	lower bound	\$4 068
Consensus level	upper bound	273

TABLE 8.30: *Approximate evaluation of objective function bounds for the higher-complexity hypothetical scenario.*

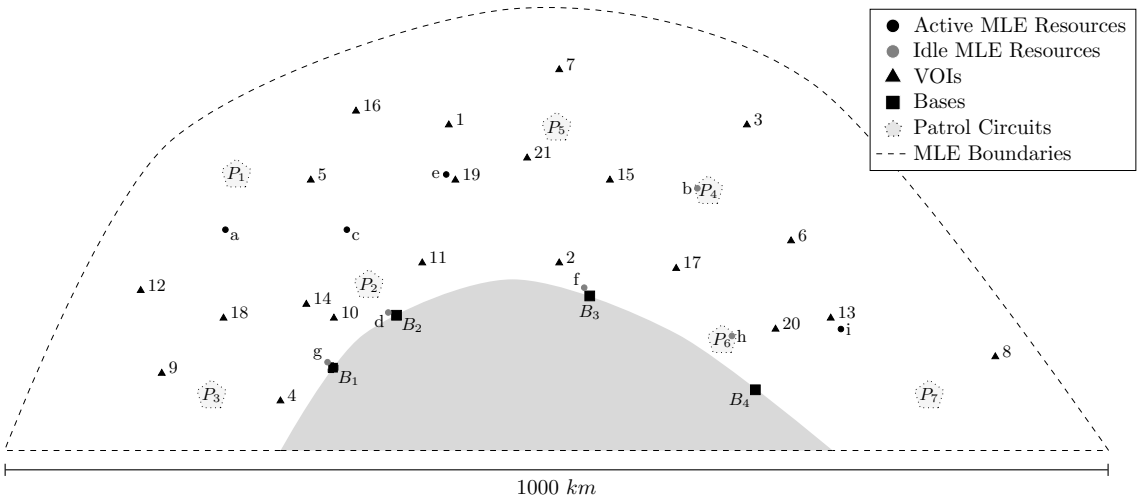


FIGURE 8.9: *Graphical representation of the physical elements in the higher-complexity hypothetical scenario.*

ideal quantities (see Table 8.28) may therefore prove to be challenging.

Furthermore, given the considerable size of this feasible search space and the very limited computation time budget available, it is expected that the search process should not waste too much effort undergoing solution exploitation procedures, but should rather attempt to find acceptable solutions from a wide variety of domain subspaces. Analogously, this means that search algorithms with higher exploration configurations are expected to perform better in a scenario of this complexity, particularly when the computation time budget is smaller.

Only the unistart and multistart archived multiobjective simulated annealing algorithms were tested in the context of this scenario, because the algorithmic structure of these search techniques excels at controlling the desired solution exploitation and exploration repartition throughout a search process (particularly with respect to inter-state subspace exploration). It is, however, duly noted that the other two search techniques may also produce good solutions to this problem instance in competition with the method of simulated annealing.

8.5.2 Simulated annealing results

In the experiments reported on in this section, the probability of performing an inter-route transformation during any given iteration is fixed to $p_{br} = 0.2$. Intra-route and inter-state transformations are, on the other hand, varied by adopting the configurations $(p_{wr}, p_{is}) = (0.6, 0.2)$, $(0.3, 0.5)$ and $(0, 0.8)$. Inter-state transformations were performed based on the structure of the binomial distribution approach of §7.5.4 with configurations $(\tilde{n}, p) = (17, 0.45)$ and $(17, 0.6)$. The steady-state probability distributions of these configurations are visually represented in Figure 8.10. Furthermore, two cooling schedules with a fixed initial temperature of 30 and a cooling ratio of 0.85, but with different numbers of epochs (400 and 700), were tested.

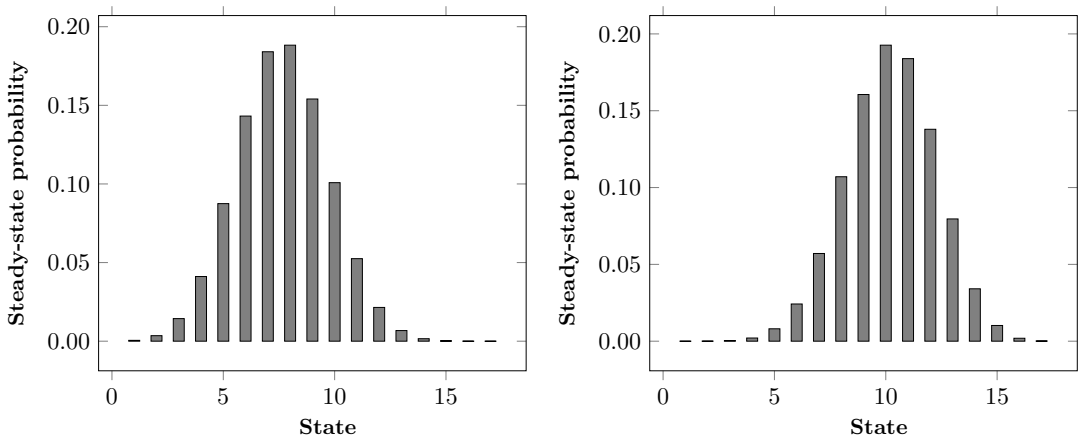
As a more suitable alternative to merely counting the number of iterations performed during a search run, it was instead decided to introduce a new sample statistic in these experimental trials representing the proportion of infeasible solutions generated during a run (namely $\overline{R}_{inf}$). This value is an indicator of computational budget waste. The various experiments performed according to the method of simulated annealing, together with the results and sample statistics obtained, are tabulated in Tables 8.31 and 8.32 for 2- and 5-minute runs, respectively.

		$(p_{wr}, p_{is}) = (0.6, 0.2)$			$(p_{wr}, p_{is}) = (0.3, 0.5)$			$(p_{wr}, p_{is}) = (0, 0.8)$		
		$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$\overline{R}_{inf}$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$\overline{R}_{inf}$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$\overline{R}_{inf}$
$(\tilde{n}, p) = (17, 0.45)$	$C = 700$	361	24 125	0.175	491	54 408	0.276	513	66 926	0.374
	$C = 400$	387	30 643	0.201	436	37 535	0.293	498	62 482	0.36
$(\tilde{n}, p) = (17, 0.6)$	$C = 700$	365	42 231	0.33	425	69 187	0.438	395	46 165	0.665
	$C = 400$	339	34 036	0.344	368	39 159	0.499	393	55 923	0.589

TABLE 8.31: *Simulated annealing results for 2-minute runs for the higher-complexity hypothetical scenario.*

		$(p_{wr}, p_{is}) = (0.2, 0.6)$			$(p_{wr}, p_{is}) = (0.3, 0.5)$			$(p_{wr}, p_{is}) = (0, 0.8)$		
		$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$\overline{R}_{inf}$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$\overline{R}_{inf}$	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$\overline{R}_{inf}$
$(\tilde{n}, p) = (17, 0.45)$	$C = 700$	515	52 442	0.192	669	99 687	0.293	703	109 243	0.406
	$C = 400$	498	50 499	0.184	591	72 590	0.301	721	114 196	0.387
$(\tilde{n}, p) = (17, 0.6)$	$C = 700$	438	75 130	0.264	499	76 707	0.499	565	113 478	0.727
	$C = 400$	477	84 437	0.298	567	103 484	0.482	571	118 095	0.65

TABLE 8.32: *Simulated annealing results for 5-minute runs for the higher-complexity hypothetical scenario.*



(a) $(\tilde{n}, p) = (17, 0.45)$ (b) $(\tilde{n}, p) = (17, 0.6)$

FIGURE 8.10: *Illustration of the steady-state probability distributions of the two inter-state binomial distribution configurations employed for the higher-complexity hypothetical scenario.*

As anticipated earlier, due to the large size of the search space and the limited computation time budget allocated, it is easy to see that experiments with higher exploration configurations irrefutably produce better results than those with lower ones. Interestingly, however, it is also apparent that a higher proportion of infeasible solutions are generated in higher-exploration experiments. This is because the method of solution exploitation using within-route transformations (see §7.5.3) generates neighbouring solutions that are close in proximity to the current solution in the feasible search space, and so the risk of producing neighbouring solutions that are outside the feasible space is significantly lowered.

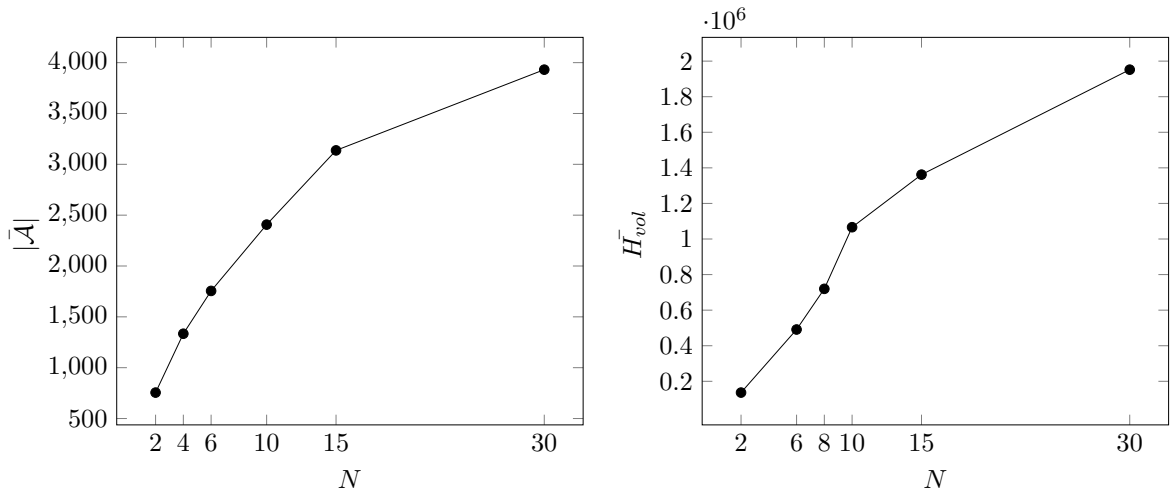
Although not discernible in the results of this section, it is also interesting to point out that the variances around the sample mean ratio of infeasible solutions observed over the course of five experiments under any given set of fixed parameters were observed to be very small generally, particularly in the case of 5-minute runs. This is again believed to a consequence of the law of large numbers. More precisely, given that each solution transformation (for a specific cooling schedule) has a fixed probability of being performed, the ratio of infeasible solutions generated by the algorithm in a given experiment will closely converge to a global mean as a result of the large number of trials (iterations) performed during the course of each search run. On the other hand, however, it should be noted that large variances were observed around the hypervolume sample means, particularly in the case of highly explorative parametric configurations.

8.5.3 Augmented multistart simulated annealing results

The various experiments performed according to the augmented multistart simulated annealing for 2-minute runs, together with the results and sample statistics obtained, are tabulated in Table 8.33. The parallel computing efficiency metrics are as defined in §8.4.5, where it is assumed that $\overline{H}_{vol}^{\mathcal{P}} \approx 2\ 100\ 000$. For the sake of clarity, results pertaining to individual search runs are this time omitted from the table, as these may be approximated with the statistics from the experimental results obtained by the unistart archived simulated annealing algorithm of the previous section. Moreover, graphical representations of the average archive sizes and hypervolume quantities observed are shown in Figure 8.11 as a function of the number of parallel processors employed.

Processors parametric configurations					Sample statistics		Performance metrics		
N	$(p_{wr}, p_{brs}, p_{brd}, p_{is})$	$(\tilde{n}, p)$	C	π	$\lvert\overline{\mathcal{A}}\rvert$	$\overline{H}_{vol}$	$\vartheta(\lvert\overline{\mathcal{A}}\rvert)$	$\vartheta(\overline{H}_{vol})$	ϑ
2	(**2**×) $(0.3, 0.1, 0.1, 0.5)$	$(17, 0.45)$	700	0.85	798	124 993	0.812	0.060	0.048
	(**2**×) $(0.3, 0.1, 0.1, 0.5)$	$(17, 0.6)$	400	0.85	675	124 747	0.917	0.059	0.054
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.45)$	700	0.85	857	136 567	0.835	0.065	0.054
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.6)$	400	0.85	691	131 333	0.880	0.063	0.055
4	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.45)$	700	0.85	1 316	317 773	0.726	0.151	0.110
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.6)$	400	0.85					
	(**2**×) $(0.3, 0.1, 0.1, 0.5)$	$(17, 0.6)$	400	0.85	1 352	333 883	0.767	0.159	0.122
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.45)$	700	0.85					
6	(**2**×) $(0.4, 0.1, 0.1, 0.4)$	$(17, 0.35)$	300	0.85	1 732	501 380	0.710	0.239	0.170
	(**2**×) $(0.4, 0.1, 0.1, 0.4)$	$(17, 0.5)$	500	0.85					
	(**2**×) $(0.4, 0.1, 0.1, 0.4)$	$(17, 0.65)$	700	0.85					
	(**2**×) $(0.2, 0.1, 0.1, 0.6)$	$(17, 0.35)$	700	0.85	1 778	481 076	0.729	0.230	0.167
	(**2**×) $(0.2, 0.1, 0.1, 0.6)$	$(17, 0.5)$	500	0.85					
	(**2**×) $(0.2, 0.1, 0.1, 0.6)$	$(17, 0.65)$	300	0.85					
8	(**2**×) $(0.4, 0.1, 0.1, 0.5)$	$(17, 0.45)$	700	0.85	1 978	719 810	0.608	0.343	0.208
	(**2**×) $(0.4, 0.1, 0.1, 0.5)$	$(17, 0.6)$	700	0.85					
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.35)$	400	0.85					
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.7)$	400	0.85					
10	(**2**×) $(0.4, 0.1, 0.1, 0.5)$	$(17, 0.45)$	700	0.85	2 407	1 066 295	0.592	0.508	0.300
	(**2**×) $(0.4, 0.1, 0.1, 0.5)$	$(17, 0.6)$	700	0.85					
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.3)$	700	0.85					
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.7)$	400	0.85					
	(**1**×) $(0.4, 0.1, 0.1, 0.2)$	$(17, 0.35)$	700	0.85					
	(**1**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.5)$	500	0.85					
15	(**3**×) $(0.4, 0.1, 0.1, 0.5)$	$(17, 0.45)$	700	0.85	3 138	1 361 900	0.514	0.649	0.334
	(**3**×) $(0.4, 0.1, 0.1, 0.5)$	$(17, 0.6)$	700	0.85					
	(**3**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.3)$	700	0.85					
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.7)$	400	0.85					
	(**2**×) $(0.4, 0.1, 0.1, 0.2)$	$(17, 0.35)$	700	0.85					
	(**2**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.5)$	500	0.85					
30	(**6**×) $(0.4, 0.1, 0.1, 0.5)$	$(17, 0.45)$	700	0.85	3 931	1 952 000	0.322	0.930	0.300
	(**6**×) $(0.4, 0.1, 0.1, 0.5)$	$(17, 0.6)$	700	0.85					
	(**6**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.3)$	700	0.85					
	(**4**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.7)$	400	0.85					
	(**4**×) $(0.4, 0.1, 0.1, 0.2)$	$(17, 0.35)$	700	0.85					
	(**4**×) $(0, 0.1, 0.1, 0.8)$	$(17, 0.5)$	500	0.85					

TABLE 8.33: *Augmented multistart archived simulated annealing results for the higher-complexity hypothetical scenario.*



(a) *Average archive sizes.* (b) *Average hypervolume quantities.*

FIGURE 8.11: *A graphical representation of some of the average archive sizes and hypervolume quantities obtained based on the number of processors employed in parallel in the augmented multistart simulated annealing algorithm for the higher-complexity hypothetical scenario.*

8.6 Post-optimisation solution analysis

As discussed previously, an acceptable non-dominated front for an MLE response selection problem instance generated during the solution search process is not only required to be in close proximity to the true Pareto front, but should also exhibit a good spread of solutions across the front and include good extremal solutions.

A visual representation of a non-dominated front generated by the solution search process of the hypothetical MLE response selection scenario of §8.3 during one of the previously conducted experimental runs is depicted in Figure 8.12. This front consists of 1 249 (non-dominated) alternatives, mapped in tetra-objective space, in which the visitation score is measured along the vertical axis, the total delay and the operating costs are measured along the horizontal axes, and the consensus level is measured by means of a varying grey shading intensity. Due to the nature of this problem, certain regions of the objective space which correspond to infeasible solutions as a result of model management restrictions are doomed to remain unexplored (particularly if short computation time budgets are enforced). This is visible in the form of gaps in objective space in the figure, which would otherwise be filled with (feasible) points in a more relaxed model formulation (or for a more relaxed computation time budget).

8.6.1 Filtering mechanisms

It is anticipated that the large number of solutions populating non-dominated sets of solutions to MLE response selection problem instances (as a result of the relatively large number of objectives) may make it difficult for the decision maker to select a suitable trade-off decision when presented with such overwhelming fronts. This is a typical case of more is not always better, as the excessive number of choices presented to an operator may require him to commit to time-consuming analyses in a limited time frame which may, in turn, cause him to select poor alternatives (from a subjectively preferential point of view). More importantly, for similar reasons to those stressed in respect of the time elapsed during the solution search process of an

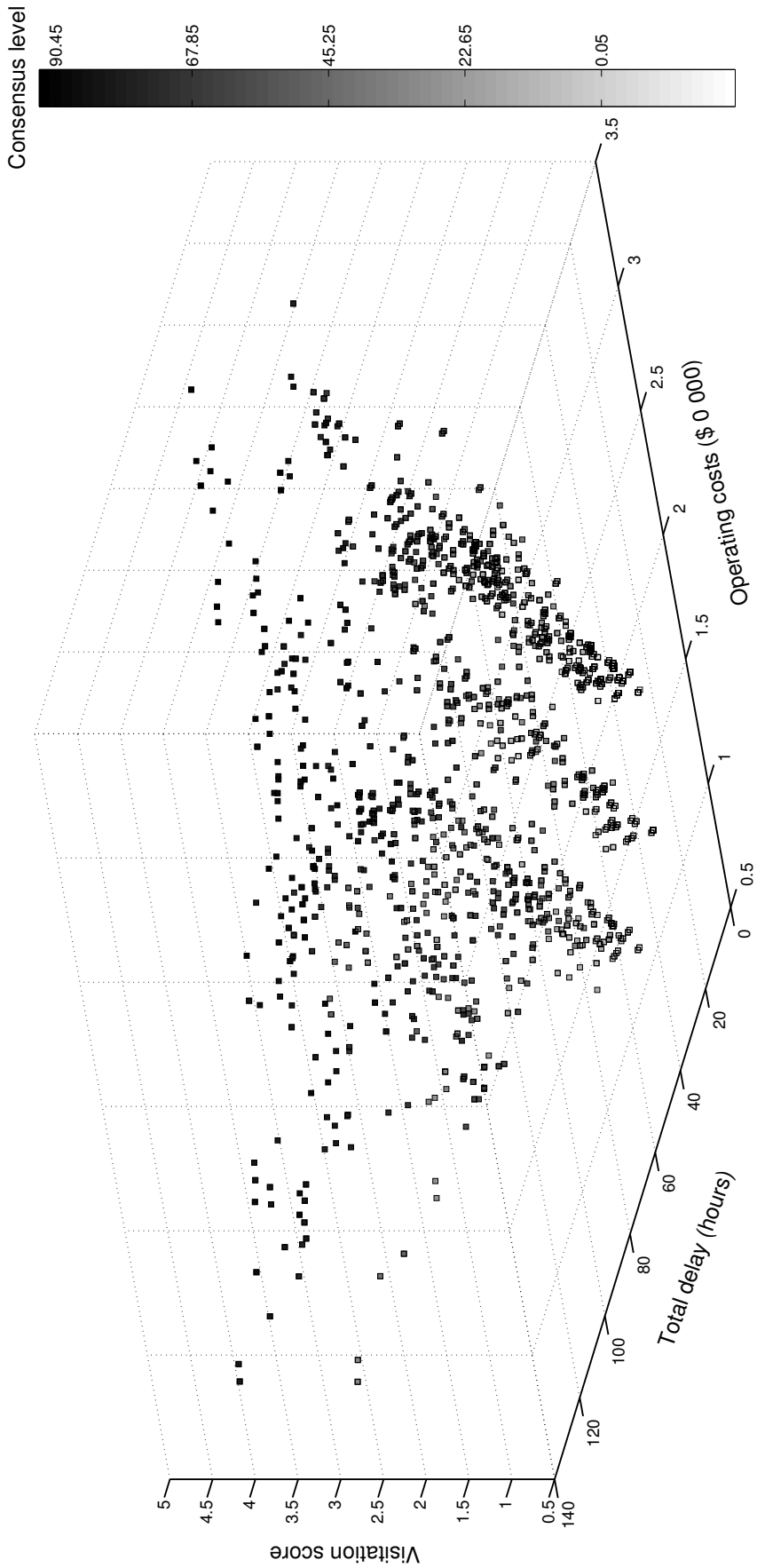


FIGURE 8.12: *A non-dominated front generated during the solution search process for the lower-complexity MLE response selection hypothetical scenario.*

MLE response selection problem instance, it is necessary for the decision maker to be able to select an alternative as rapidly as possible in order to prevent the deterioration of MLE response selection operations caused by late changes in solution implementation.

It is therefore advocated that one or more filtering mechanisms be employed to reduce the number of solutions in the non-dominated front *a posteriori* of the solution search process. An example of basic filtering techniques may include the use of *objective function bounds*, where certain thresholds are specified in order to dismiss certain subsets of points in objective space (and, of course, the corresponding points in solution space) which violate these thresholds.

To illustrate this filtering technique, consider the following (hypothetical) situation. As a result of an insufficient budget allocation towards MLE operations, suppose that the operator does not wish to select alternatives that are expected to cost more than \$20 000. Additionally, suppose that the behaviour of more threatening VOIs detected within the jurisdiction area of the hypothetical coastal nation is often unpredictable and intelligent, to the extent that they have in the past often managed to escape from the tracking of threat detection systems as a result of late servicing. The operator may wish to remedy to this problem by ignoring solutions that generate excessively large delay scores, and decide to impose a total delay upper bound score of 100 hours. Lastly, in order to remain on good terms with the decentralised MLE response selection decision making entities with whom he collaborates, as well as to take into account their expert advice in respect of which detected VOIs they reckon they are competent to investigate, the (central) operator may decide only to consider solutions associated with a consensus level of at least 40 points. The result of the above-mentioned impositions placed on the non-dominated front of Figure 8.12 is portrayed in Figure 8.13. For visual purposes, note that the spectrum of grey intensities quantifying the fourth objective has been rescaled.

Here, it is interesting to point out that there exists a strong correlation between the visitation score and the consensus level (*i.e.* the higher the solutions are mapped on the visitation score axis, the darker they seem to be). This is because the cumulated ideal quantity of VOIs of the hypothetical MLE response selection scenario of §8.3 amounts to 10, and so solutions residing in higher state subspaces are generally preferred by the decision making entities (that is, provided that the correct assignment of VOIs to entities is performed).

The implementation of these objective bounds allowed for the original non-dominated set to be reduced by over 50%, rendering the subsequent selection process easier. Of course, the stricter these objective bounds are, the more reduced the non-dominated set will be. However, even after the use of this kind of filter, there may still be too many alternatives. A more sophisticated procedure is the ϵ-*filter* technique [167], which aims to partition areas of high solution density by slicing objective function ranges of values at regular intervals in order to form M-orthotopes in the M-dimensional objective space. A single non-dominated point, located in each orthotope, is then retained, provided that there is at least one point in the orthotope, while the remaining points are removed from the front and solution space.

To better describe this filtering process, consider Figure 8.14 in which a non-dominated front has been generated for solving a bi-objective minimisation problem with objective functions f_1 and f_2. Here, the parameters ϵ_1 and ϵ_2 are used to partition the objective function ranges at regular intervals into regular intervals of lengths ϵ_1 and ϵ_2, respectively, and discarding solutions so that only one point in each of the resulting cross-sectional rectangles remains. Of course, the larger the parameter values ϵ_1 and ϵ_2 are, the smaller the remaining number of non-dominated solutions will be. In particular, it is noted that $|\tilde{\mathcal{F}}^{\epsilon}| \to |\tilde{\mathcal{F}}|$ as $\epsilon_1, \ldots, \epsilon_M \to 0$ (where $\tilde{\mathcal{F}}^{\epsilon}$ represents the resulting ϵ-filtered non-dominated front).

To illustrate the functionality of the above-mentioned filtering mechanism in a more complex

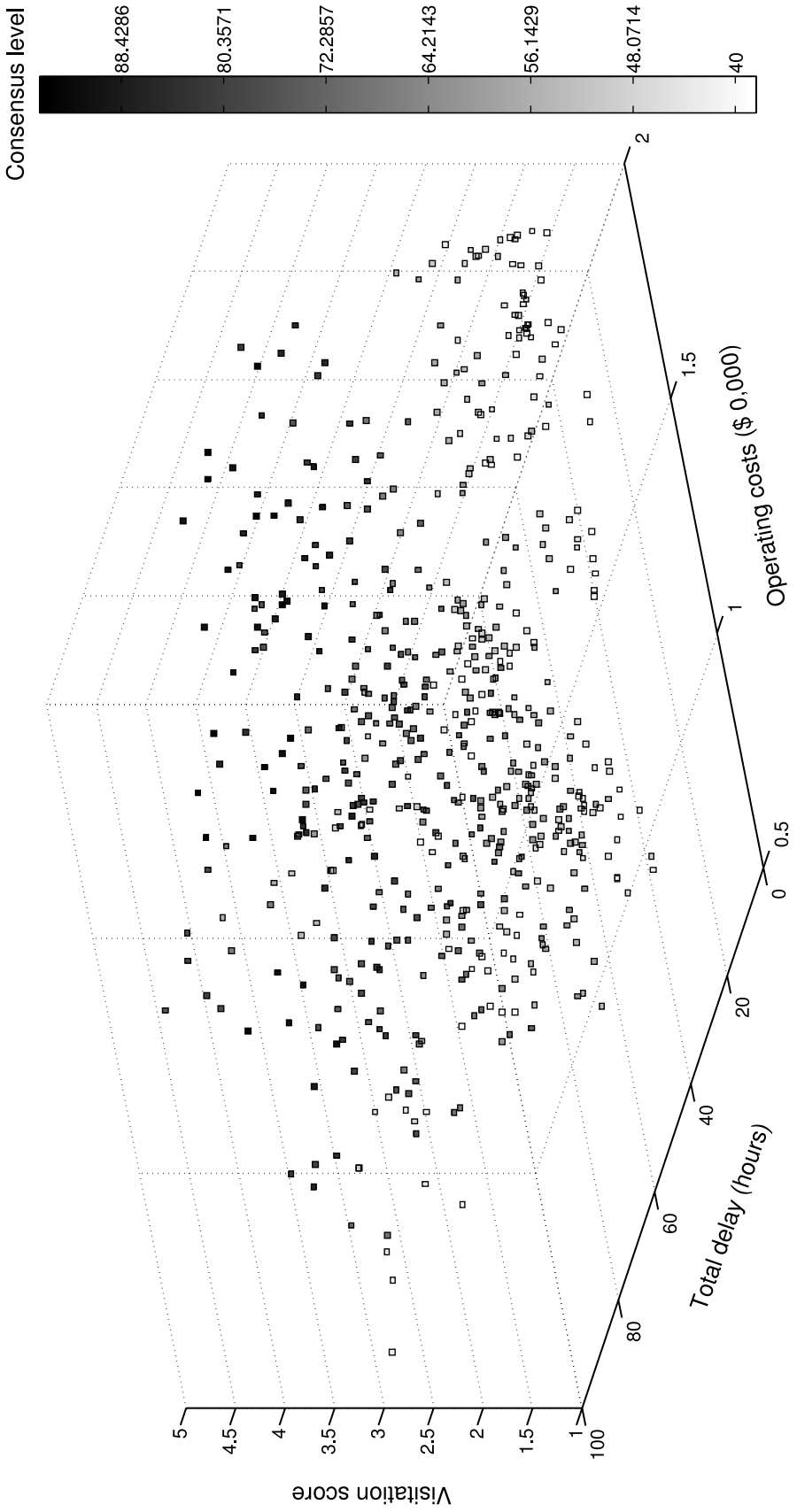
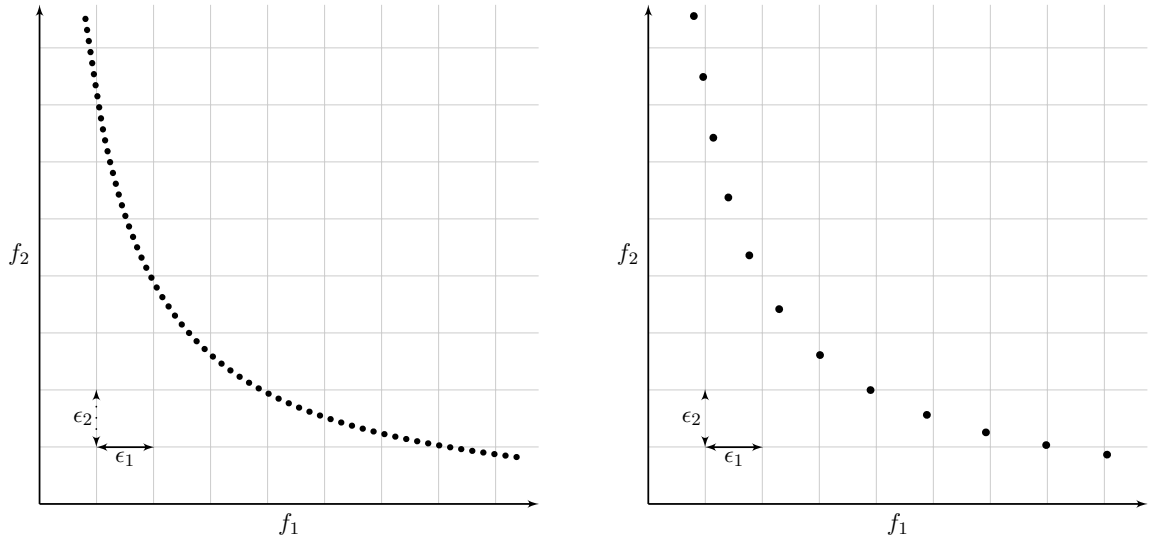


FIGURE 8.13: *Non-dominated front filtered from Figure 8.12 by implementing post-optimisation objective function bounds for the lower-complexity hypothetical scenario.*



(a) *Non-dominated front before implementing the ϵ-filter technique.*

(b) *Non-dominated front after implementing the ϵ-filter technique.*

FIGURE 8.14: *Application of the ϵ-filter technique to the non-dominated front of a bi-objective minimisation problem with objective functions f_1 and f_2.*

objective space environment, reconsider the (reduced) non-dominated front of Figure 8.13, and suppose that the *ϵ-filter* technique is applied with ϵ-values $(\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4) \approx (1, 25, 500, 12.5)$. The resulting (reduced) non-dominated front generated as a result of this process is shown in Figure 8.15. Here, each objective function range was sliced exactly three times, creating $4^4 = 256$ 4-orthotopes across the objective space. Although certain clusters of points may seem too "close" to one another in respect of the three-dimensional Euclidean objective space, the reader is reminded that, in such cases, the closeness of these points differ largely in their values of the fourth objective function (*i.e.* significantly different shades of grey are observed amongst these points).

The filtered non-dominated front obtained by applying these two filtering techniques in combination produces a total of fifty one non-dominated solutions, which clearly is a significant improvement over the initial non-dominated front (over twenty four times smaller than the initial number of non-dominated solutions). For the sake of clarity, these points may also be projected orthogonally from three-dimensional Euclidean objective space (*i.e.* considering the first three objectives only) onto the three two-dimensional planes, as shown in Figure 8.16.

Although not portrayed here, the more advanced *cluster filtering* techniques may also be employed by grouping non-dominated points into clusters in such a way that any two alternatives in a specific cluster are sufficiently "similar" to one another while being sufficiently "different" to the alternatives included in other clusters [73, 110]. One representative in each cluster (*e.g.* its *centroid*) may then be retained, representing the characteristics of the cluster as a whole, while the other points in that cluster are discarded. In [110], for instance, a filtering technique referred to as *subspace clustering* is employed, which is an extension of traditional clustering that seeks to find clusters in different subspaces within a data set made up of highly-dimensional observations (such as the set of non-dominated solutions mapped in objective space in an MLE response selection problem).

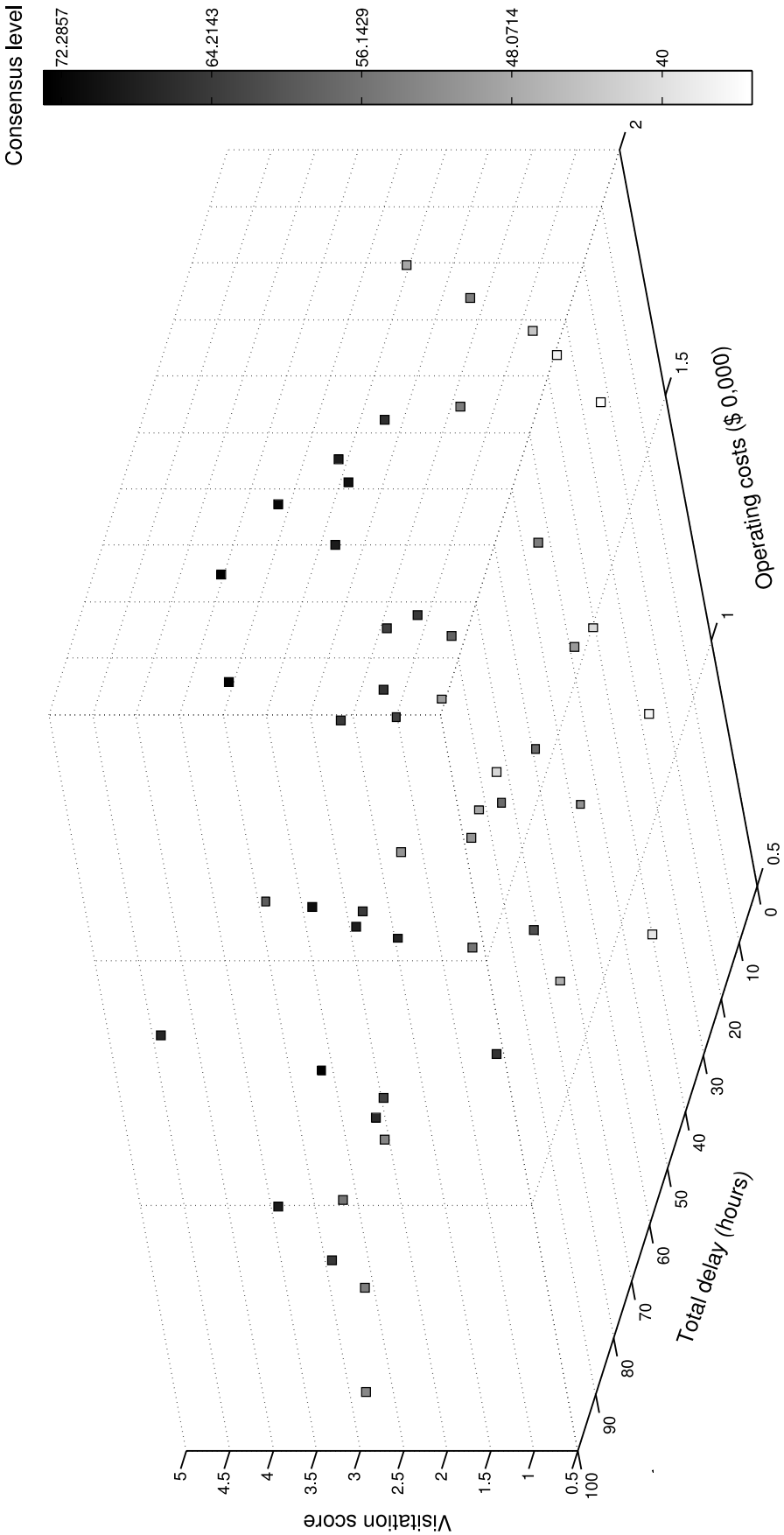


FIGURE 8.15: *Non-dominated front filtered from Figure 8.13 by implementing the ϵ-filter for the lower-complexity hypothetical scenario.*

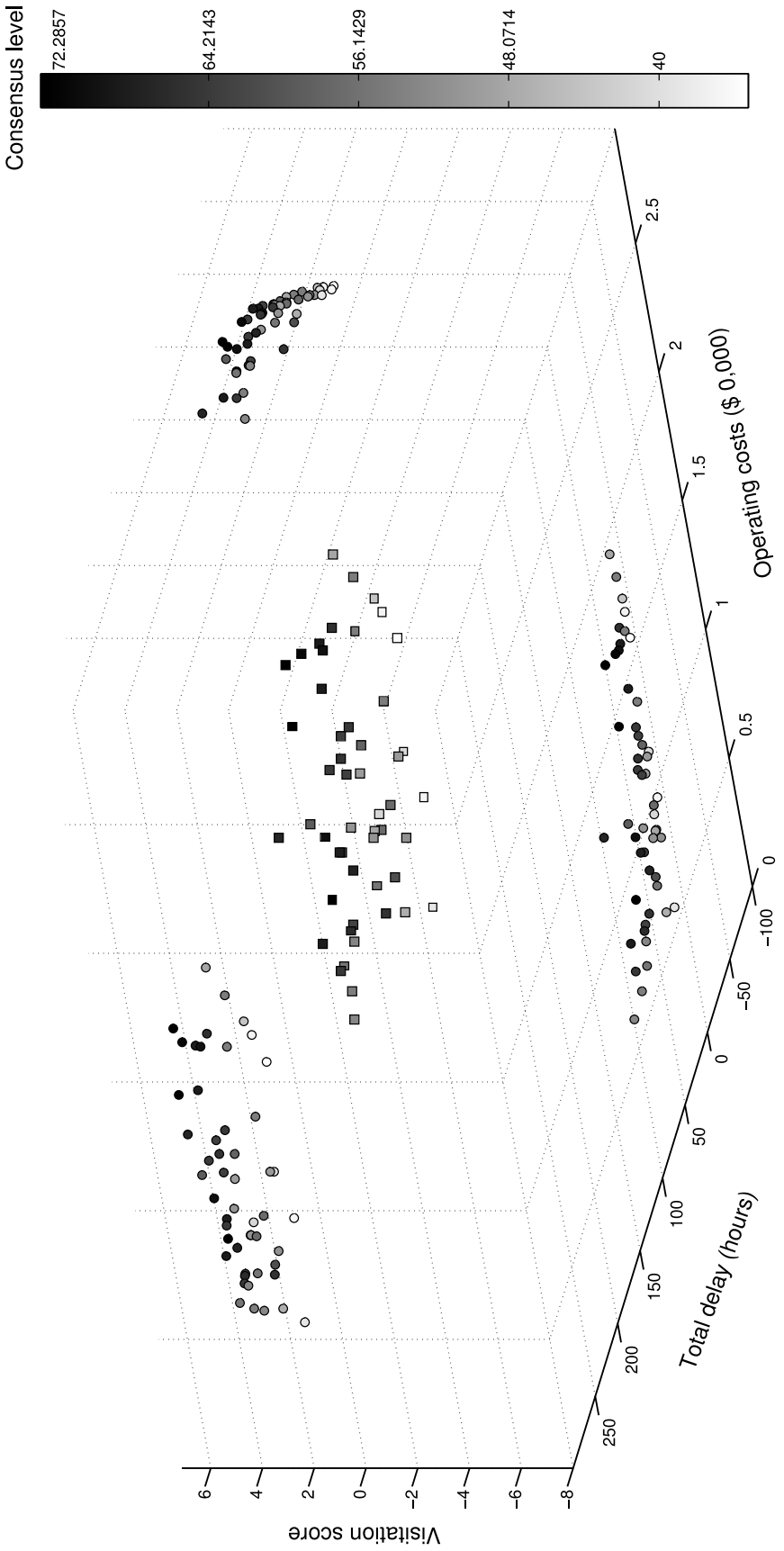


FIGURE 8.16: *Non-dominated front reduced from Figure 8.13 together with orthogonal projections of this front onto three 2-dimensional planes for the lower-complexity hypothetical scenario.*

$\mathbf{0}_a$	9	P_3	$\mathbf{0}_b$	7	3	P_4	$\mathbf{0}_c$	10	5	B_2	$\mathbf{0}_e$	1	P_5	$\mathbf{0}_f$	2	11	B_2	$\mathbf{0}_g$	4	B_1	$\mathbf{0}_h$	6	B_4

FIGURE 8.17: *String representation of a candidate solution $\boldsymbol{x}$ to the MLE response selection problem instance for the lower-complexity hypothetical scenario which results in the objective function vector $f(\boldsymbol{x}) = (4.28, 39.88, 1.49, 83.3)$.*

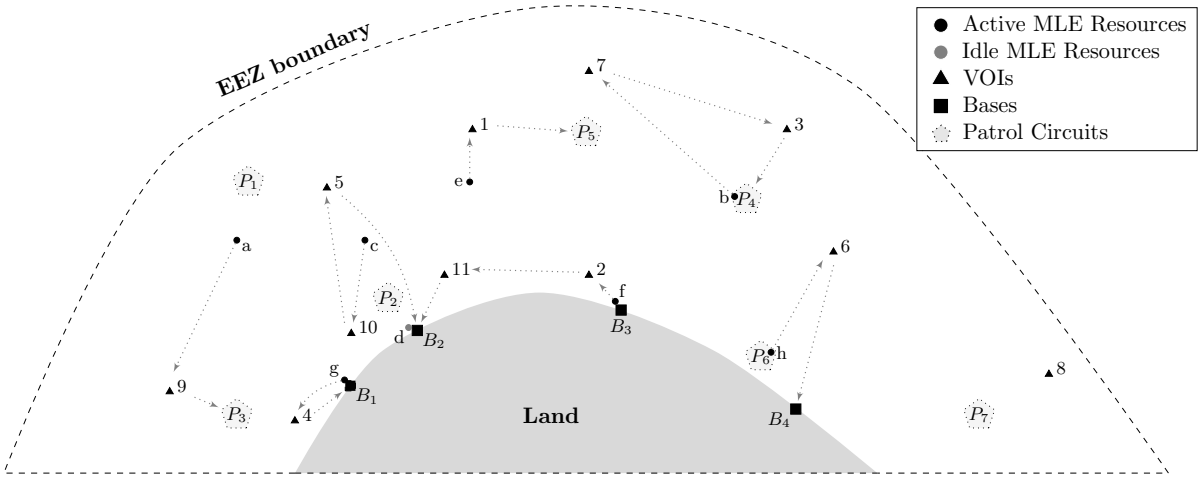


FIGURE 8.18: *Graphical representation of the solution in Figure 8.17.*

8.6.2 Alternative selection

The reduced non-dominated front in Figure 8.16 may be made available to the MLE response selection operator who is then required to select a single, most preferred compromise solution. The selected point in objective space may then be mapped back to its associated solution (or one of its associated solutions, in cases of surjective mapping) in solution space. This solution may finally be communicated to the MLE database, from where the various MLE response selection decision making entities and other various role-players are immediately notified of the changes in routing implementations brought along by this new solution.

Suppose that the alternative induced by the vector $f(\boldsymbol{x}) = (4.28, 39.88, 1.49, 83.3)$ in this objective space is deemed to be the most preferable by the operator. The associated solution is shown (in string form) in Figure 8.17, while the associated configuration of its interception routes scheduled at the beginning of the time stage is shown in Figure 8.18.

8.7 Chapter summary

The aim in this chapter was to validate the proposed MLE response selection DSS of Chapter 3 by:

1. Motivating the use of an adequate stochastic multiobjective optimisation performance assessment quality indicator,

2. Creating two complete hypothetical MLE response selection scenarios of varying complexities using the parameters and concepts introduced in Chapters 4 and 5,

3. Proposing a mathematical model considering various major features of the MLE response selection problem induced by the hypothetical scenarios in (2),
4. Conducting a series of experiments using various combinations of parametric configurations in conjunction with various search techniques for solving the mathematical model in (3),
5. Commenting on the performance levels of the results obtained in (4), and
6. Describing the purpose and functionality of the post-optimisation component.

To this effect, a review of various means of assessing the performances of multiobjective stochastic optimisation techniques was conducted in §8.1, with particular focus on the hypervolume quality measure. This was followed by an evaluation of the domain size of the MLE response selection problem in §8.2. A lower-complexity MLE response selection hypothetical scenario, together with a representative mathematical model for this scenario, were then put forward in §8.3 and a number of numerical experiments were performed in §8.4, where the results of these experiments were reported and analysed in detail. These experiments were then repeated in §8.5 in the context of a higher-complexity hypothetical scenario using a subset of the proposed solution search techniques applied in §8.4. Finally, a description of the post-optimisation solution analysis component of the DSS proposed in this dissertation was provided in §8.6, using various graphical illustrations.

Although two hypothetical MLE response selection scenarios of varying complexities were presented and analysed in this chapter, it is acknowledged that real-life scenarios will continually vary in complexity and search space geometry. The computational experiments conducted in this chapter were therefore not aimed at finding the best possible algorithm and parametric configuration for the scenarios under consideration, but rather to gain some insight into the working of the proposed solution search techniques, from where relevant and useful observations could be made in respect of their potential role in the implementation of a real-life MLE response selection DSS. Although the experiments performed in this chapter were deemed successful, in reality they may still fall short in quality if an MLE response selection operator expects optimisation processes to run for no longer than computation time budgets that are significantly smaller than the ones employed in these experiments.

In the next chapter, a dynamic multi-level framework is proposed in which the synchronised use of real-time system memory is combined with three DSS dynamism components in order to (1) enhance the execution of the proposed problem instance reinitialisation methodology as disturbances occur in the picture unfolding at sea and (2) provide optimisation techniques at the beginning of time stages with *a priori* information on the nature of the current decision and objective spaces as a means to boost the performance of the solution search processes.

CHAPTER 9

Consolidation of system dynamism

Contents

As explained in the previous two chapters, one of the major complications associated with the on-line approach (see §2.7.1) adopted to solve the MLE response selection problem of this dissertation relates to the restriction imposed by very limited computational budget time on the expectations that the optimisation procedures will find good solutions in very large search spaces at the start of randomly occurring new problem instances. The primary goal in this penultimate chapter is to analyse the various dynamic constituents of the proposed MLE response selection DSS and to consolidate them into a functional framework as a means to solve the MLE response selection problem on a time continuum in an effective and realistic manner. In particular, the working of the information management subsystem components are presented in detail in this chapter.

This chapter is structured as follows. The problem instance reinitialisation component is first presented in detail in §9.1, and this is followed by a theoretical discussion on so-called problem instance landscape shifts in the context of MLE response selection in §9.2. A technique employed in preventing the deterioration of solutions generated during specific response selection time windows is then introduced in §9.3, while the need for jump-start optimisation in solving the on-line MLE response selection problem is motivated in §9.4. A full description of the solution tracking and optimisation tracker components of the DSS proposed in this dissertation may be found in §9.5 and §9.6, respectively, accompanied by a proposed dynamic on-line optimisation framework in §9.7. The focus then turns to a discussion on the real time micro-management of MLE resources in §9.8, after which the chapter closes with a brief summary in §9.9.

9.1 The problem instance reinitialisation component

In §4.1.5, the notion of *time stages* was introduced in order to accommodate the dynamic nature of the MLE response selection problem. According to this definition, an MLE response selection environment is subject to disturbances, which may be described as threshold phenomena occurring stochastically in time and causing the currently implemented non-dominated solution to suffer significantly in terms of performance, possibly to such an extent that it no longer satisfies the inter-objective preferential trade-off values of the operator, or becomes dominated.

In order for the MLE response selection DSS to acknowledge the start of a new time stage, disturbances may basically either be induced manually or automatically, as a result of a significant change in one or more (simultaneous) input data streams. While a human operator may easily be made aware of certain types of disturbances, other types of disturbances are not so obviously detectable, and this may cause the operator to dither on the decision as to whether or not to trigger a new time stage. In addition, in situations where real-time input data are overwhelming in terms of quantity or complexity, it may be difficult for the operator to identify these input changes on his own. In particular, the operator or his aids may not always be fixated on the DSS screens, analysing the data, but might be busy with other tasks. As a result, delays may arise between the time at which a significant change in input data occurs and the operator's realisation that this change has taken place.

Consequently, the main purpose of the *problem instance reinitialisation* component, introduced in §3.1.3, is to detect threshold violations in input data streams and to alert the DSS or operator of such occurrences. In this section, the constituent elements of this component are presented. Examples of disturbances spanning all three MLE response selection decision making paradigms are described, and this is followed by computational examples of two automated instance reinitialisation protocols.

9.1.1 Input data threshold violations

A disturbance in the MLE response selection problem formulation is typically caused as a result of one of the following occurrences:

1. Detection of a new VOI,
2. Disappearance of an unserviced VOI scheduled for visitation,
3. Significant changes between the actual and anticipated velocity vector of a VOI,
4. A significant difference between the actual and expected interception point with a VOI,

5. Significant updates with respect to the expected threat nature of a VOI,
6. A significant difference between the actual and expected servicing time of a VOI,
7. An infeasible encounter,
8. A violation of capacity restrictions,
9. The availability of a previously unavailable MLE resource,
10. The unexpected unavailability of a currently active MLE resource,
11. Significant changes observed in external environmental vectors,
12. Significant changes reported in the VOI preference ordered set of a decision entity, or
13. An operator-induced event.

In the above list, Disturbance 1 refers to input data being made available to the operator in respect of a newly detected VOI in the jurisdiction area of the coastal nation, while Disturbance 2 refers to a VOI that has already previously been detected (and assigned to an MLE resource visitation route) but which is lost by the detection systems before being intercepted (this may be caused by radar system malfunction or a large accumulated delay time associated with visiting that VOI).

Disturbance 3 simply refers to a VOI that experiences significant and unexpected changes in its current velocity so that the calculation of new travel trajectories and, possibly, the re-allocation of certain MLE resources is necessary. Similarly, Disturbance 4 occurs in cases where the actual interception point of an MLE resource with a VOI deviates significantly from its expected interception point, which may cause solution deterioration if this MLE resource is expected to service further VOIs afterwards due to possible additional travel distances. Disturbance 5 refers to a significant update in respect of the potentially threatening nature associated with a certain VOI, possibly causing the re-allocation of a new MLE resource to that VOI, and hence interfering with some of the currently implemented routes. Disturbance 6 pertains to the time taken to investigate/neutralise a VOI, which may deviate significantly from its expected service time, and may hence potentially interfere with the currently implemented solution.

Disturbance 7 refers to the case where an infeasible encounter occurs between an MLE resource and a VOI, as described in §4.1.6. Disturbance 8 is an exception in respect of Observation 6 of §4.1.2; it occurs when an MLE resource is obliged to return to a base after servicing one or more VOIs along its route embodying one or more types of incompatible threats, similar to the notion of a capacity constraint being violated in traditional VRPs. Disturbance 9 refers to an MLE resource that was previously unavailable (*e.g.* the MLE resource was not idle, was undergoing repair or maintenance, or was in the process of re-supplying for its following mission once it had relocated to a base at the end of its previous mission) but which is now ready to be assigned to a route (following the usual set-up procedures of active MLE resources). On the other hand, Disturbance 10 takes cognisance of the fact that an MLE resource currently in use may suddenly become unavailable (such as when it breaks down).

Disturbance 11 refers to changes in the nature of environmental factors, particularly oceanic currents and weather conditions, which are important factors in the accurate calculation of interception trajectories, and may possibly interfere with the set of routes in the current solution. Disturbance 12 acknowledges that significant changes in preferential subsets of VOIs from the decision entities may force the central operator to reconsider the current distribution of VOIs

amongst the decision entities (in the decentralised decision making paradigm), or by designing new, and possibly better, sets of routes accommodating such changes (in an intermediate decision making paradigm). It is acknowledged that such disturbances will typically occur as a result of input data change observations, therefore rendering them redundant in cases where these are triggered following changes in input data triggered alongside any other disturbances. Finally, Disturbance 13 refers to situations triggered by the operator himself in respect of conducting certain strategic manoeuvres at sea, or purely because he intuitively believes that the system should search for a new solution. Engaging in a long distance chase with an uncooperative VOI at the moment of interception is an example of such a disturbance.

9.1.2 Automated instance reinitialisation protocol examples

Although the description of detecting significant input data threshold violations by means of disturbances makes theoretical sense, the question remains, however, as to what extent of data violations are deemed significant enough to warrant resolving the MLE response selection problem instance entirely or in part. While certain cases of input data changes may trigger obvious disturbances (consider, for example, Disturbances 1, 2 and 9 in the list of §9.1.1), others may not be as trivial and may require the configuration of complex metrics and threshold calculations. To illustrate the potential working of two such input data violations, examples of advanced automated instance reinitialisation protocols, namely those pertaining to Disturbances 3 and 5 in the list of §9.1.1, are presented next.

Threshold violations in VOI anticipated velocity vectors using Gaussian corridors

In this example, various stochastic constituents derived in Chapter 6 are employed as a means to establish threshold violations in anticipated VOI velocity vectors. Recall from §6.3.7 that the concept of confidence ellipses may be thought of as bounding a measurable set of random variables distributed around the central tendencies of the underlying bivariate Gaussian probability distribution. It was also discussed how these ellipses may be employed as a means to model the uncertainty surrounding the anticipated locations of VOIs, derived from their anticipated velocity vectors. In addition, recall from §6.3.6 that the uncertainty surrounding the trajectory derived from the anticipated velocity vector of a VOI may be expressed as a continuous function of time by employing the dynamic bivariate Gaussian probability distribution. It follows that these confidence ellipses may thus also be defined as a continuous function of time.

It was furthermore demonstrated in §6.3.4 how the probability domain of VOI locations could be mapped from a temporary set of local axes onto a global frame of reference by means of a rotation transformation

$$\begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} = \begin{bmatrix} \cos\theta_i(t) & -\sin\theta_i(t) \\ \sin\theta_i(t) & \cos\theta_i(t) \end{bmatrix} \begin{bmatrix} \tilde{x}_i(t) \\ \tilde{y}_i(t) \end{bmatrix}.$$

It was also mentioned in §6.3.7 that the major and minor diameter lengths of a confidence ellipse mapped locally, with an amplitude under-shaped by an $\alpha\%$ confidence threshold, are given by $2\breve{\sigma}_{iX}\sqrt{\chi^2_\alpha(2)}$ and $2\breve{\sigma}_{iY}\sqrt{\chi^2_\alpha(2)}$, respectively. Recalling that the major axis of a confidence ellipse is always aligned with the x-axis of its corresponding local frame of reference, it follows that the intersection points between the major and minor axes and the confidence ellipse with respect to

the global frame of reference are given by the two-dimensional coordinates

$$\left(\boldsymbol{P}^{-}_{maj}(x_i), \boldsymbol{P}^{-}_{maj}(y_i)\right)(t) = \begin{bmatrix}\mu_{iX}(t)\\ \mu_{iY}(t)\end{bmatrix} - \begin{bmatrix}\cos\theta_i(t) & -\sin\theta_i(t)\\ \sin\theta_i(t) & \cos\theta_i(t)\end{bmatrix}\begin{bmatrix}\breve{\sigma}_{iX}(t)\sqrt{\chi^2_\alpha(2)}\\ 0\end{bmatrix}$$
$$= \begin{bmatrix}\mu_{iX}(t) - \breve{\sigma}_{iX}(t)\sqrt{\chi^2_\alpha(2)}\cos\theta_i(t)\\ \mu_{iY}(t) - \breve{\sigma}_{iX}(t)\sqrt{\chi^2_\alpha(2)}\sin\theta_i(t)\end{bmatrix}$$

and

$$\left(\boldsymbol{P}^{+}_{maj}(x_i), \boldsymbol{P}^{+}_{maj}(y_i)\right)(t) = \begin{bmatrix}\mu_{iX}(t) + \breve{\sigma}_{iX}(t)\sqrt{\chi^2_\alpha(2)}\cos\theta_i(t)\\ \mu_{iY}(t) + \breve{\sigma}_{iX}(t)\sqrt{\chi^2_\alpha(2)}\sin\theta_i(t)\end{bmatrix}$$

along the major axis of the confidence ellipse, and (similarly)

$$\left(\boldsymbol{P}^{-}_{min}(x_i), \boldsymbol{P}^{-}_{min}(y_i)\right)(t) = \begin{bmatrix}\mu_{iX}(t)\\ \mu_{iY}(t)\end{bmatrix} - \begin{bmatrix}\cos\theta_i(t) & -\sin\theta_i(t)\\ \sin\theta_i(t) & \cos\theta_i(t)\end{bmatrix}\begin{bmatrix}0\\ \breve{\sigma}_{iY}(t)\sqrt{\chi^2_\alpha(2)}\end{bmatrix}$$
$$= \begin{bmatrix}\mu_{iX}(t) + \breve{\sigma}_{iY}(t)\sqrt{\chi^2_\alpha(2)}\sin\theta_i(t)\\ \mu_{iY}(t) - \breve{\sigma}_{iY}(t)\sqrt{\chi^2_\alpha(2)}\cos\theta_i(t)\end{bmatrix}.$$

and

$$\left(\boldsymbol{P}^{+}_{min}(x_i), \boldsymbol{P}^{+}_{min}(y_i)\right)(t) = \begin{bmatrix}\mu_{iX}(t) - \breve{\sigma}_{iY}(t)\sqrt{\chi^2_\alpha(2)}\sin\theta_i(t)\\ \mu_{iY}(t) + \breve{\sigma}_{iY}(t)\sqrt{\chi^2_\alpha(2)}\cos\theta_i(t)\end{bmatrix}.$$

along the minor axis of the confidence ellipse. A subset of these coordinates are shown in Figure 9.1.

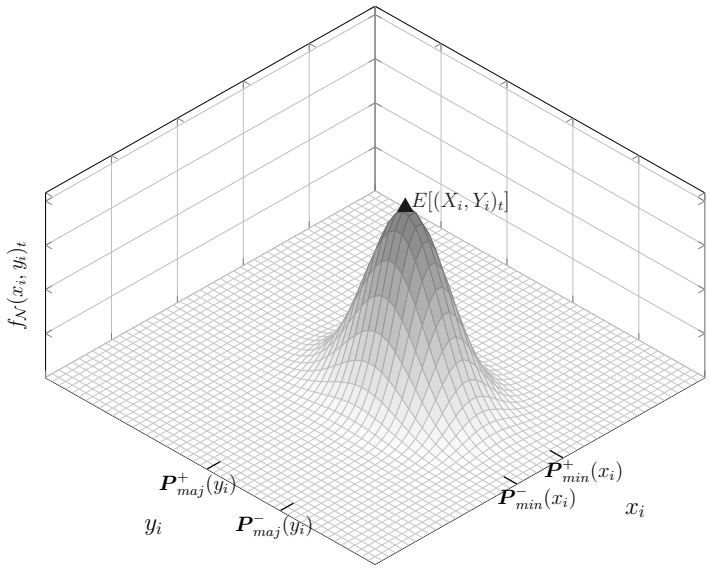


FIGURE 9.1: *Intersection coordinates between the diameters and contour of a confidence ellipse.*

The dynamic bivariate Gaussian probability distribution modelling the anticipated trajectory of a VOI may be represented visually by super-positioning an infinite number of consecutive bivariate Gaussian distributions over a vector trajectory consisting of anticipated central tendencies mapped in a bounded space-time continuum. To this effect, define the *Gaussian corridor* $\underline{\overline{\boldsymbol{G}}}$ of a VOI as the space-time mapping of the dynamic bivariate Gaussian distribution of its anticipated locations over a specific time interval. This corridor may be employed to model the displacement of a Gaussian random variable governed by a dynamic bivariate Gaussian distribution as a function of time. Moreover, define the α-*bounded* Gaussian corridor $\underline{\overline{\boldsymbol{G}}}(\alpha, i)$ of VOI i as the

space-time mapping of the confidence ellipses derived from the underlying distribution over a specific time interval. The space enclosed by these boundaries is then defined by the confidence ellipse contour coordinates (derived above) over that time interval.

This notion is best demonstrated by means of a simple example. Consider some VOI i at the beginning of a time stage with estimated initial position $(X_i, Y_i)_0 = (50, 60)$, static anticipated velocity vector (see §6.3.2) $\tilde{\boldsymbol{v}}_i = [\theta_i \; s_i] = [\frac{\pi}{12} \; 10]$, and corresponding anticipated location vector

$$\tilde{\boldsymbol{\ell}}_i(t) = \left(50 + 10t\cos\frac{\pi}{12}, 60 + 10t\sin\frac{\pi}{12}\right) \approx (50 + 9.7t, 60 + 2.6t).$$

Moreover, suppose that the error associated with the anticipated location of this VOI as a function of time is estimated by means of the variance functions $\breve{\sigma}^2_{iX}(t) = 0.2 + 1.5t$ and $\breve{\sigma}^2_{iY}(t) = 0.15 + 0.8t$ (see §6.3.4), and assume that temporal projections on the anticipated locations of this VOI are only required up to three hours from time $t = 0$. Suppose further that the operator wishes to configure the Gaussian corridor of this VOI in such a way that there is no more than an estimated 10% chance that the actual trajectory of this VOI will exit this corridor at any point during the course of the following three hours. Noting that $\chi^2_{90}(2) \approx 4.61$ and recalling that the vector of central tendencies $\boldsymbol{\mu}_i(t)$ is presumably given by $\tilde{\boldsymbol{\ell}}_i(t)$, the boundaries of this Gaussian corridor along the minor diameters of the confidence ellipses are therefore approximated by the set of parametric equations

$$\begin{aligned}
\overline{\underline{\boldsymbol{G}}}(\alpha, i)^-_{min} &= \left(\boldsymbol{P}^-_{min}(x_i), \boldsymbol{P}^-_{min}(y_i)\right)(t) \\
&= \begin{bmatrix} 50 + 9.7t + (0.15 + 0.8t)\sqrt{4.61}\sin\frac{\pi}{12} \\ 60 + 2.6t - (0.15 + 0.8t)\sqrt{4.61}\cos\frac{\pi}{12} \end{bmatrix} \\
&\approx \begin{bmatrix} 50.16 + 10.56t \\ 59.72 + 1.11t \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
\overline{\underline{\boldsymbol{G}}}(\alpha, i)^+_{min} &= \left(\boldsymbol{P}^+_{min}(x_i), \boldsymbol{P}^+_{min}(y_i)\right)(t) \\
&= \begin{bmatrix} 50 + 9.7t - (0.15 + 0.8t)\sqrt{4.61}\sin\frac{\pi}{12} \\ 60 + 2.6t + (0.15 + 0.8t)\sqrt{4.61}\cos\frac{\pi}{12} \end{bmatrix} \\
&\approx \begin{bmatrix} 49.84 + 8.84t \\ 60.16 + 3.46t \end{bmatrix}
\end{aligned}$$

for all $t \in [0, 3]$. These equations may alternatively be expressed in terms of the global frame of reference as $\overline{\underline{\boldsymbol{G}}}(\alpha, i)^-_{min} : y_i \approx 0.11x_i + 54.42$ and $\overline{\underline{\boldsymbol{G}}}(\alpha, i)^+_{min} : y_i \approx 0.39x_i + 40.65$ for all $x_i \in [50, 70.1]$. The set of parametric equations $\overline{\underline{\boldsymbol{G}}}(\alpha, i)^-_{maj}$ and $\overline{\underline{\boldsymbol{G}}}(\alpha, i)^+_{maj}$ may be derived similarly.

An approximate graphical depiction of this Gaussian corridor is shown in Figure 9.2 and its spatial projection is elucidated in Figure 9.3. A case of disturbance is also illustrated in Figure 9.2, where the actual trajectory of VOI i, $\boldsymbol{\ell}_i(t)$, exits the Gaussian corridor at time $t = 2.4$ hours approximately.

Although this example represents a simple case of an anticipated VOI trajectory, which allows for an easy visual representation of its corresponding Gaussian corridor, it is noted that the underlying geometrical shape of hyper-dimensional Gaussian corridors becomes rather tedious when dealing with non-linear velocity vector parametric equations. Finally, it is noted that α-bounded Gaussian corridors need not only be derived from a continuous bivariate Gaussian distribution. If only a time series of discrete Gaussian distributions is employed instead (see §6.3.4), then the corresponding Gaussian corridor may simply be extrapolated approximately from the edges of the underlying (discrete) confidence ellipses.

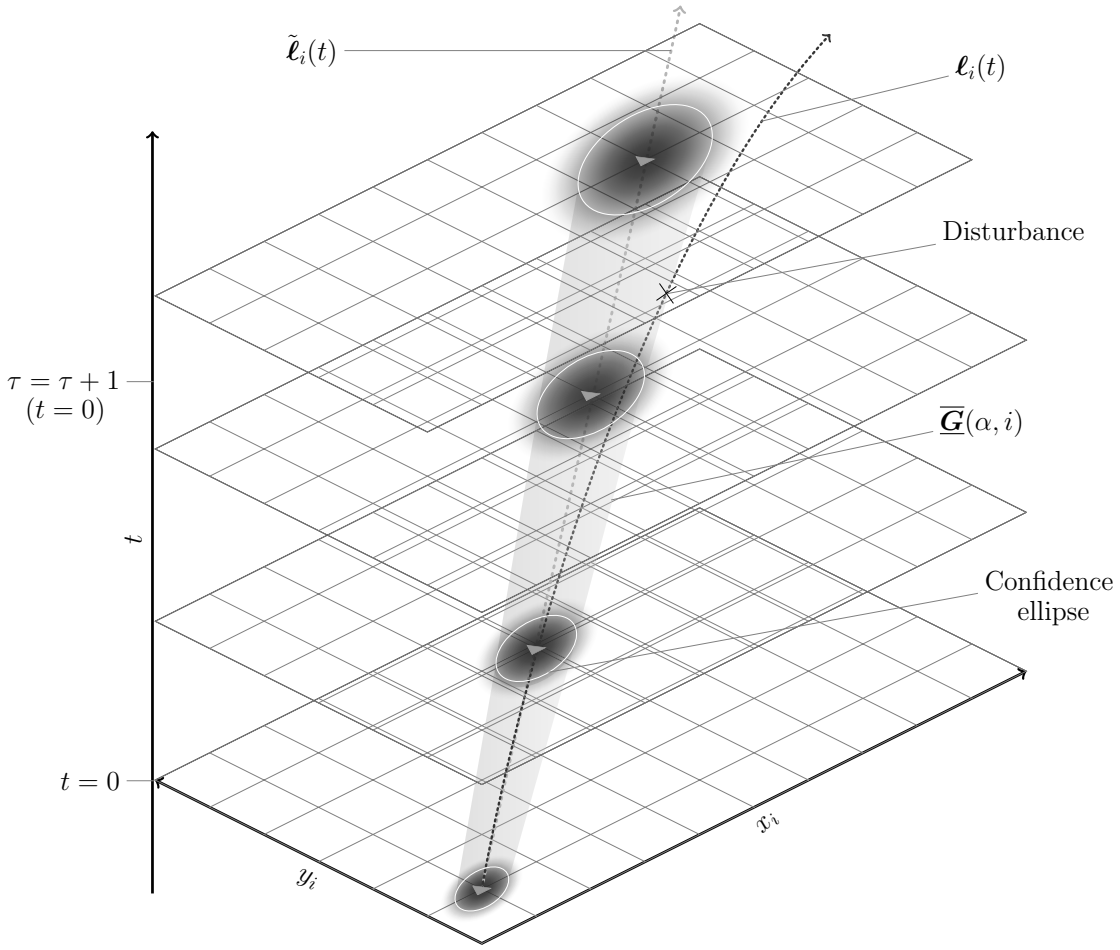


FIGURE 9.2: *Hypothetical space-time representation of the α-bounded Gaussian corridor of VOI i.*

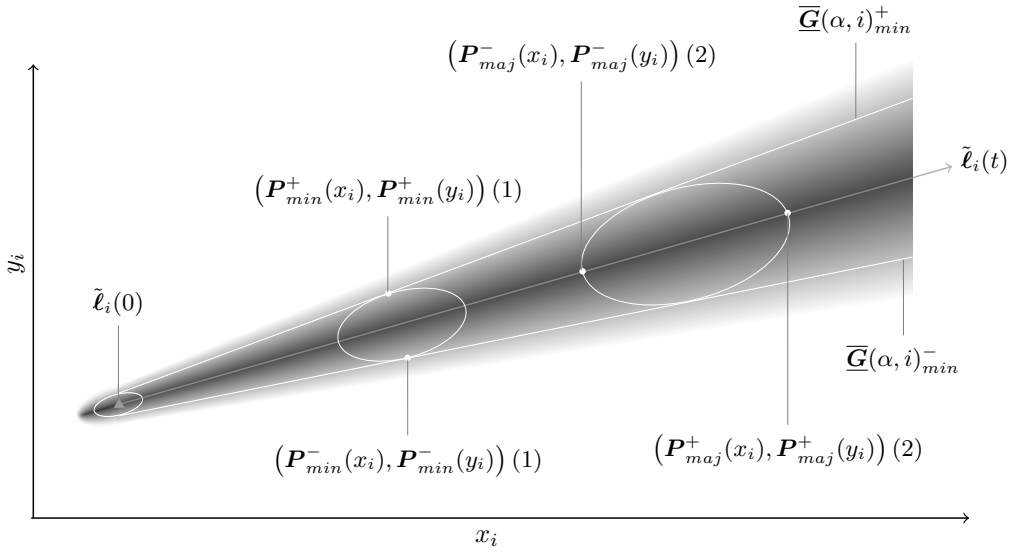


FIGURE 9.3: *Tri-dimensional spatial projection of the graph in Figure 9.2.*

Threshold violations in VOI threat probabilities using the weighted p-norm

Disturbances caused by significant updates of the threat probability vector of a VOI may cause a scheduling problem for the MLE resource assigned to intercept it. If $\mathcal{H}$ is the set of indices representing each type of threat faced by the coastal nation (as defined in Chapter 4), then let $\boldsymbol{p}_i^t = [\boldsymbol{p}_{i1}^t, \ldots, \boldsymbol{p}_{i|\mathcal{H}|}^t]$ be the threat probability vector of VOI $i \in \mathcal{V}_\tau^e$ at time $t \in \Re$, and define the *weighted p-norm* of the difference between the threat probability vector of VOI i at the beginning of the current time stage and at present time as

$$\left|\left|\boldsymbol{p}_i^0 - \boldsymbol{p}_i^{t^*}\right|\right|_p = \left(\sum_{h\in\mathcal{H}} Q_h \left|\boldsymbol{p}_{ih}^0 - \boldsymbol{p}_{ih}^{t^*}\right|^p\right)^{\frac{1}{p}},$$

where $p \geq 1$ (by definition of the standard p-norm), $t^* > 0$ and Q_h represents the priority level assigned by the coastal nation to neutralise class h threats (as described in §4.2.1). The output given by the proposed weighted p-norm is therefore more sensitive to changes in probabilities associated with threats subject to higher priority levels. Henceforth, define $\Delta\boldsymbol{p} \in (0, 1)$ as a pre-determined threshold set by the operator in the sense that the subsystem is configured to trigger a disturbance (and immediately inform the operator in real time) if there exists a VOI $i \in \mathcal{V}_\tau^e$ for which $\left|\left|\boldsymbol{p}_i^0 - \boldsymbol{p}_i^{t^*}\right|\right|_p > \Delta\boldsymbol{p}$ at any time t^*.

9.2 Problem instance landscape shifts

In dynamic on-line optimisation problems, specific deviations in input data ought to affect the geometry of the decision and objective spaces in certain ways. It may thus be beneficial to understand how these spaces tend to transition as a result of these changes for the purpose of, for example, selecting adequate problem instance reinitialisation optimisation methods best able to re-track a shifted Pareto front. This metamorphosis of the problem instance nature is referred to here as *landscape shifts*.

Because every problem instance is different and every change in input data is unique, however, it may not be practical to analyse specific problem instance landscape shifts every time these manifest themselves. Instead, the aim in this section is to put forward a generic classification of problem instance landscape shifts that may be employed in multiobjective dynamic on-line optimisation problems, and to elucidate these categories with examples of input data variations that may occur within the MLE response selection DSS proposed in this dissertation.

Problem landscape shift category	1	2	3	4	5	6
The previous feasible decision space remains entirely feasible and valid?	✓	✓	✓	×	×	×
None of the previous solutions are affected in objective space?	✓	×	×	✓	×	×
Some of the previous solutions are affected across all objectives at once?	×	×	✓	×	×	✓

TABLE 9.1: *General categories of multiobjective optimisation landscape shifts (× = no; ✓= yes).*

Six categories of MLE response selection landscape shifts are considered in this section; these are summarised in Table 9.1. Here, a solution to an MLE response selection problem instance is defined to be *invalid* if it no longer forms part of the domain space ($\mathbb{S}$) of the instance; that is, if it is no longer applicable to the current situation. For the sake of illustration, a centralised MLE response selection decision making paradigm involving Objectives I (visitation score), II (total

delay) and III (operating costs) of Chapter 4 is assumed in this section. In addition, only input data changes associated with the event of the disturbance triggers listed in §9.1 are considered.

9.2.1 Category 1 shifts

As described in Table 9.1, *Category 1* shifts retain all individual solutions during a transition from the previous feasible search space to the new one, while simultaneously retaining all non-dominated solutions from the previous problem instance. In other words, landscape shifts of this kind occur whenever changes in input parameters have no impact on any objective or constraint functions. A figure illustrating a hypothetical example of a Category 1 landscape shift for a dynamic multiobjective problem with two objective functions f_1 and f_2 (which are to be minimised) is shown in Figure 9.4. In the event of such a landscape shift, it is of course suggested that the problem instance reinitialisation optimisation process be configured to primarily seek good solutions within the new problem instance subspace.

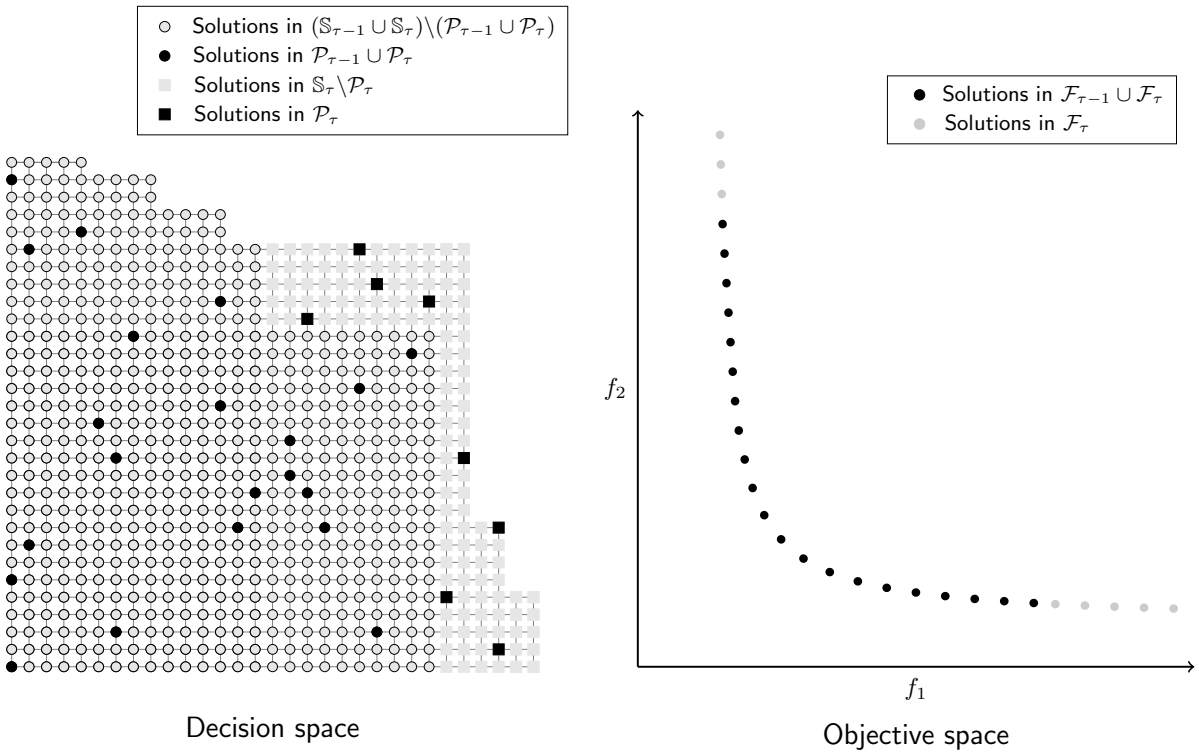


FIGURE 9.4: *Illustration of the effect of a Category 1 shift on the decision and objective spaces of a hypothetical bi-objective problem instance.*

Within the context of this MLE response selection problem, disturbances caused by the *detection of a new VOI* is an example of a Category 1 shift, where the feasible search space of the previous time stage becomes a subspace of the new feasible search space. Because the triangle inequality does not always hold in the MLE response selection problem, it is acknowledged that inserting the newly detected VOI into a route may result in shorter travelling distance and, hence, shorter operating costs. It is, however, also acknowledged that the total delay of a given route will never be reduced if an additional VOI is added to that route. Moreover, the visitation score of a solution will generally increase if an additional VOI is inserted into its underlying string encoding. A previously non-dominated solution will therefore never become dominated in the event of such a disturbance. Because the new domain subspace region is one that exclusively

encloses all solutions containing a route visiting the newly detected VOI, it follows that any of the new solutions will never improve across all objectives simultaneously.

9.2.2 Category 2 shifts

In the case of *Category 2* shifts, all feasible solutions from the previous problem instance will remain feasible in the new decision space, but some of the previously non-dominated solutions may become dominated in the new objective space. Additionally, some of the new feasible solutions may also become non-dominated during the new time stage. Landscape shifts of this kind therefore occur whenever changes in input data have no impact on the constraint functions, but affect a certain subset (but not all) of the objective functions. An example of a Category 2 shift for the same hypothetical problem as in Figure 9.4 is illustrated in Figure 9.5.

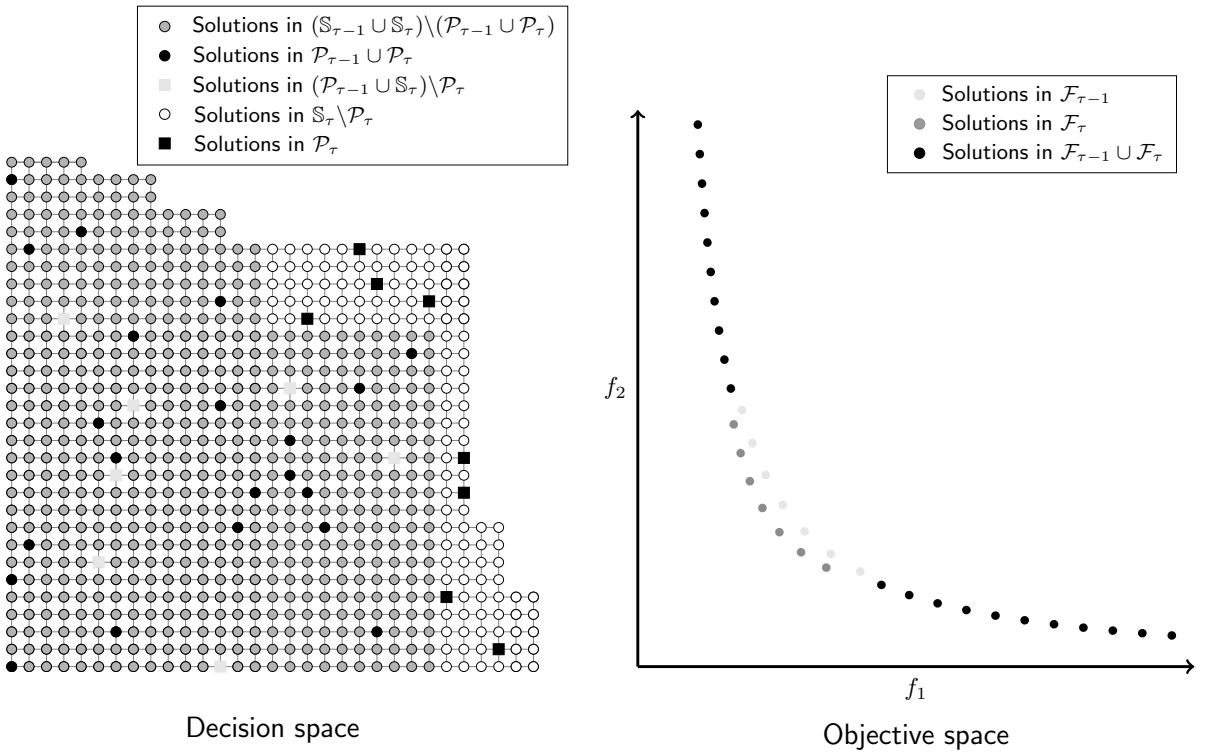


FIGURE 9.5: *Illustration of the effect of a Category 2 shift on the decision and objective spaces of a hypothetical bi-objective problem instance.*

Disturbances caused by *previously unavailable MLE resource becoming available* is an example of a Category 2 shift. In this situation, all solutions belonging to the feasible search space of the previous time stage form a subspace of the new feasible domain, containing all solutions that do not assign any VOIs to the newly available MLE resource (*i.e.* solutions in which this MLE resource is inactive). Because allocating a specific subset of VOIs in a solution to the route associated with this new MLE resource, in contrast to allocating them to one or more less suited MLE resources, may improve the quality of the solution across all objective functions, it follows that some of the previously non-dominated solutions may consequently become dominated. In addition, because all feasible solutions from the previous time stage remain feasible, it follows that all previously dominated solutions will remain dominated during the new time stage (transitive properties of domination).

If an intermediate decision making paradigm is implemented, then disturbances caused by *changes in VOI preference vectors* also fall within the class of Category 2 shifts. In this case, however, the decision space actually remains identical, while only the consensus score is affected in objective space. Solutions in objective space will thus merely relocate to some new point along the consensus score axis only (whilst maintaining all other objective scores). Any non-dominated solution from the previous time stage for which there exists at least one other non-dominated solution vector that has better scores in the first three objectives is therefore a candidate to becoming dominated during the new time stage.

9.2.3 Category 3 shifts

Category 3 shifts are similar to Category 2 shifts, with the exception that certain previously dominated solutions remaining feasible in the new search space have the possibility of becoming non-dominated in the new (feasible) objective space, noting that dominated solutions cannot shift to a non-dominated status unless they are able to improve across all objective functions and/or if the solution(s) dominating it become(s) infeasible and/or invalid. Landscape shifts of this kind are therefore caused whenever changes in input parameters have no impact on any constraint functions, but affect all objective functions at once.

Disturbances caused by a *significant update in the threat probability vector of a VOI* may be classified as a Category 3 shift. Here, changes in the threat probabilities of a VOI will not impact the feasible domain (under the fundamental model constraints[1]), but will have a direct impact on the visitation score. In addition, this will also indirectly affect servicing time, in turn impacting both the operating costs and the total delay time as well. Consequently, certain solutions that were previously dominated may become non-dominated in the new problem instance as they have the opportunity to improve in respect of all three objective functions.

9.2.4 Category 4 shifts

Unlike the previous three landscape shift categories, *Category 4* shifts involve changes in the feasible decision space of a problem instance, so that solutions that were previously feasible may no longer be so. In this category, however, none of the feasible solutions remaining from the previous problem instance are affected in objective space.

Disturbances caused by a *currently active MLE resource becoming unavailable* is an example of a Category 4 shift (which is the reverse situation of a Category 2 shift for *previously unavailable MLE resource becoming available* described earlier). In this type of situation, all solutions that scheduled the concerned MLE resource to visit one or more VOIs (*i.e.* solutions in which this MLE resource was in an active state) are scrapped from the feasible decision space and labelled as invalid. The new search space is therefore significantly shrunk, so that the feasible subset of solutions in which this MLE resource was in an idle state in the previous problem instance now entirely spans the new decision space. Meanwhile, the new objective space mappings are not affected, but because invalid solutions are also discarded from the objective space, some of the feasible solutions that were previously dominated by these solutions may hence now become non-dominated.

Disturbances caused by the *disappearance of a VOI scheduled for visitation* also belong to this category. In such events, the feasible search space of the previous time stage is also shrunk

[1]Cases in which a change in threat probabilities results in infeasible assignments, causing certain solutions no longer to be feasible would, however, qualify as a Category 6 shift.

significantly, as all solutions that scheduled the concerned VOI to be visited by any active MLE resource are rendered invalid, and hence not considered in the new search space. Again, the remaining solutions remain unaffected in both decision and objective spaces, some of which may possibly transition to a non-dominated status.

9.2.5 Category 5 shifts

As shown in Table 9.1, the difference between Category 4 and *Category 5* landscape shifts is that some of the feasible solutions of the new decision space are shifted across a certain subset (but not all) of the objective functions, so that new non-dominated solutions may be generated otherwise than by merely discarding invalid solutions that previously dominated them.

Disturbances caused by a *longer than expected servicing time*, for example, belong to the class of Category 5 shifts. In this scenario, the delay and operating costs are directly affected as new interception trajectories have to be recalculated for most solutions. It is therefore possible that the affected route, in addition to certain other configuration of routes involving this particular VOI, may violate the autonomy constraints, in turn possibly telescoping certain non-dominated solutions from the previous feasible objective space outside of the new feasible objective space. The visitation score, however, remains unaffected here. Previously dominated solutions may therefore only become non-dominated in the new problem instance if the solution(s) previously dominating them have become infeasible. Disturbances caused by *the actual interception point with a VOI differing significantly from its expected interception point* and by *significant changes in observed environmental vectors* may also be classified as Category 5 shifts for similar reasons.

9.2.6 Category 6 shifts

Category 6 shits are the most perturbing ones, causing a subspace of solutions to be affected both in decision space and across all objectives simultaneously. Disturbances caused by *infeasible encounters* or by certain *operator-induced events* are examples of problem instance landscape shifts belonging to this category.

9.3 Temporal retraction of foreseen dynamic input parameters

Based on the various constituent elements of the mathematical modelling components developed previously in this dissertation, it is noted that one way of classifying the various model input parameters[2] is to regard them as either fixed in the medium to long term or typically evolving in the short to medium term. It is suggested that dynamic input parameters may furthermore be classified according to both the degree of frequency of their transition through time and the extent of anticipated knowledge with respect to the future nature of these transitions. A classification of this type of the foremost dynamic input parameters of this MLE response selection DSS is provided in Table 9.2. Here, the bold entries indicate the presence of dynamic input parameters transitioning in a quasi or semi-continual fashion.

While the nature of the *unknown* parameters as a function of time may, by definition (see §2.6.2), not be anticipated, the nature of the *stochastic* parameters may, to varying extents (see Chapter 6), be projected into the future with the use of probabilistic modelling techniques. Furthermore,

[2] A distinction is made between *input* and *computed* parameters, where the former are received from sources that are external to the MLE response selection DSS, while the latter are derived from these input parameters in the mathematical modelling subsystem of the DSS (see Figure 6.1 for examples of computed parameters).

Deterministic	Stochastic	Unknown
VOIs accumulated delays	**VOIs current locations**	VOIs threat probabilities
MLE resources current locations	**VOIs anticipated trajectories**	VOI preference ordered sets
MLE resources trajectories	Environmental conditions	
MLE resources autonomy levels		

TABLE 9.2: *Classification of the foremost MLE response selection DSS dynamic input parameters with respect to transition frequency and predictability.*

the nature of the deterministic dynamic input parameters may be predicted with near certainty in the short term. Here, a dynamic input parameter is referred to as *foreseen* if it exhibits either deterministic or stochastic characteristics.

Although somewhat straight forward, the on-line optimisation procedure adopted in the MLE response selection DSS proposed in this dissertation may, as mentioned previously, be significantly compromised on multiple levels due to the duration of response selection time windows. One such restriction concerns the so-called input data *snapshot effect* (see §3.2), in which dynamic model parameters are frozen in time for the entire duration of the response selection time window. This may lead to inaccurate situation representation as the dynamic input parameters will, meanwhile, have evolved in real time during the course of that response selection time window, in turn causing the chosen solution implementation following the termination of the optimisation process no longer to be a true representation of reality (to a certain extent). This is particularly pronounced in cases where the response selection time window durations are typically long, in cases where certain dynamic input parameters are transitioning at a high rate of change and/or in cases where the underlying problem landscape is typically very sensitive to changes in input parameters.

Because an algorithmic time-limit stopping criterion is employed during the course of optimisation procedures, and assuming that the model management and post-optimisation processes are generally carried out quickly and/or over a fixed period of time, it is possible for the duration of the response selection time window to be closely known in advance. It is therefore proposed that the issue pertaining to inaccurate solution representation be corrected accordingly by retracting *a priori* the numerical values from frequently transitioning foreseen dynamic input parameters, calculated or estimated from the end of the duration of the response selection time window to the beginning of the time stage.

In other words, it is advocated that a more representative mathematical model ought to (*inter alia*) incorporate numerical parameters that are applicable to the MLE situation at the time of the solution implementation, and not at the time of the disturbance. This concept is illustrated in Figure 9.6 (not to scale), and examples of basic information retraction computations involving certain deterministic dynamic input parameters from Table 9.2 are shown in Table 9.3, where the function domain t is defined in *minutes*.

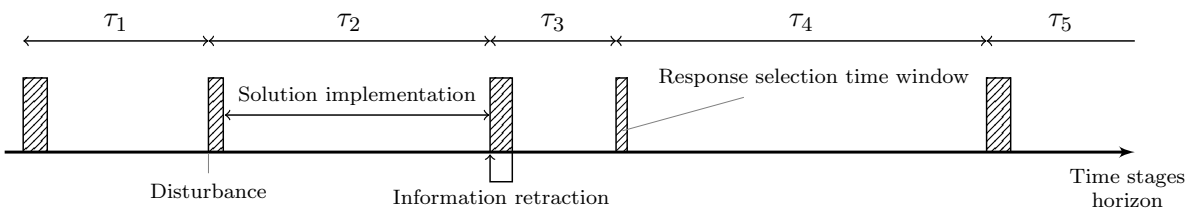


FIGURE 9.6: *Generic representation of the time stages horizon in the MLE response selection DSS.*

Parameter	Present value	Transition function	Retracted value
VOI i accumulated delay	$T_{i\tau}(0) = 45$	$T_{i\tau}(t) = 45 + t$	$T_{i\tau}(5) = 50$
MLE resource k location	$0_{k\tau}(0) = (230, 45)$	$0_{k\tau}(t) = (230 + 0.25t, 45 - 0.3t)$	$0_{k\tau}(5) = (231.25, 43.5)$
MLE resource k distance autonomy level	$a^d_{k\tau}(0) = 1100$	$a^d_{k\tau}(t) = 1100 - 0.4t$	$a^d_{k\tau}(5) = 1098$

TABLE 9.3: *Numerical examples of temporal retraction of foreseen dynamic input parameters for hypothetical transition functions and estimated five-minute response selection time windows.*

9.4 Jump-start optimisation for MLE response selection

As discussed in §2.7.2, it has been demonstrated in the literature that significant computational effort may often be saved, and better solution quality achieved, by reusing information gained in past problem instances whilst solving on-line, dynamic multiobjective optimisation problems. This computational advantage was referred to as *jump-start optimisation*.

Besides being an on-line, dynamic multiobjective optimisation problem itself, it is suggested, that the MLE response selection process could immensely benefit from jump-starting a response selection solution search process. Indeed, based on the experiments conducted in Chapter 8, starting the search anew may prove to be too impractical in real-life situations when the response selection time window is narrow, the size of the search space is very large, and disturbances occur too frequently. It is therefore believed that the best hope of achieving effective MLE response selection reinitialisation procedures may involve tracking the set of Pareto-optimal solutions as closely as possible both within and outside response selection time windows.

Investing effort in jump-starting optimisation procedures is, however, only justified if two consecutive problem instance landscapes are sufficiently similar and, more importantly, if the set of carried-over, non-dominated solutions (*i.e.* the earliest estimate of the Pareto-optimal set of solutions of the new problem instance) meets a certain standard with respect to quantity, quality and diversity. Past a certain point the effort may be considered superfluous, in which case it would be at least as efficient to restart the search process anew using no *a priori* information. Moreover, it is important to acknowledge that the magnitude of certain input data variations are not necessarily directly proportional to the magnitude of the shifts in problem instance landscapes. A small, local change in decision space may, for instance, have a significant impact on the shift and shape of the Pareto front in objective space. Fortunately, this is a much more serious concern in single-objective dynamic optimisation problems, where the (only) optimum may relocate to a completely different, perhaps distant region of the domain space, rendering information associated with the previous optimum useless. The tetra-objective nature of the MLE response selection problem investigated in this dissertation is, however, expected to contain a very large number of Pareto-optimal solutions (see Chapter 8), in turn improving the chances that a significant proportion of these solutions will be retained over any two consecutive problem landscapes.

The proportion and nature of Pareto-optimal solutions retained will, of course, significantly be influenced by the type of problem landscape shift that has taken place, and it is acknowledged that the intensity of these perturbations may also be affected by the degree of input data variation that has taken place. Moreover, although only disturbances were considered as examples of events causing these shifts, it is critical to acknowledge that problem landscape shifts are expected to occur almost continuously, independently from the advent of disturbances.

In other words, numerous consecutive "small" but accumulating landscape shifts are actually bound to occur between any two problem instance reinitialisations, and so tracking the shift in Pareto-optimal solutions may not be as simple as suggested in §9.2, but may rather be a compounded accumulation of many landscape shifts of different categories and different intensities. An approximated Pareto optimal set may consequently not be useful in jump-start optimisation procedures if it is not tracked efficiently, particularly if input data variations are generally frequent and disturbances are far apart.

Generic jump-starting techniques for dynamic multiobjective VRPs are, however, very scarce in the literature, and most related studies only tend to give examples for simplified, generic mathematical models. In agreement with Cámara *et al.* [20], the author consequently speculates that there exists no common, unified structure for solving the like of the MLE response selection problem in a dynamic fashion. In the following three sections, the solution tracking and optimisation tracker components are therefore presented in some detail with the aim of associatively structuring a functional jump-starting optimisation framework for use in problem instance reinitialisation procedures.

9.5 The solution tracking component

In order to carry over a chosen set of solutions from one time stage to the next, it is necessary to continually evolve these solutions in both decision and objective space. This process is referred to here as *solution tracking*, which takes place in the solution tracking component of the information management subsystem introduced in §3.1.3.

The solutions to be tracked need not only consist of a current estimate of the Pareto-optimal set $\tilde{\mathcal{P}}_\tau$ (which, of course, also contains the currently chosen solution for implementation), but may also consist of a mutually exclusive set of strategically (or randomly) retained dominated and infeasible solutions, defined here as $\mathcal{D}_\tau$. Examples of elements that may be appended to this set include extreme solutions, solutions promoting diversity, isolated solutions in objective space, solutions belonging to second or third non-dominated fronts, and infeasible solutions closely located to feasible domain subspaces of interest in decision space.

9.5.1 The solution tracking algorithm

Solution tracking is a simple, low complexity process, which works as follows. Updates in input data variations that have ensued since the start of the previous solution tracking time interval, referred to here as a *time split*, are first received from external data sources at the start of the current time split (if any[3]). Unlike the conditions necessary to trigger disturbances (as discussed in §9.1), however, these input data updates account for *any* changes in problem formulation, no matter how small (such as the accurate monitoring of the updated geographical positions of VOIs and MLE resources). Solutions that are both in $\tilde{\mathcal{P}}_\tau$ and $\mathcal{D}_\tau$ are first assessed for invalidity (recalling that a solution is classified as *invalid* if it does not form part of the current problem domain). Consider, for example, solutions scheduled to visit a VOI which no longer belongs to the set $\mathcal{V}_\tau^e$ after it was serviced during the course of the current time stage.

Next, solutions in $\tilde{\mathcal{P}}_\tau$ are reassessed for feasibility in decision space, taking into consideration the possible domain landscape shifts that have transpired since the start of the previous time split. Based on the chosen configuration, infeasible solutions in $\tilde{\mathcal{P}}_\tau$ are then either transferred to

[3] In cases where the duration of a time split is quicker than the rate of input data updates, for example.

the set $\mathcal{D}_\tau$ or discarded from the solution tracking process. The remaining (feasible) solutions in $\tilde{\mathcal{P}}_\tau$ are then reassessed in objective space. Following this, all feasible solutions in $\mathcal{D}_\tau$ are also reevaluated in objective space, and a temporary set $\mathcal{D}_{\tau\succ}$ containing all feasible, non-dominated solutions in $\mathcal{D}_\tau$, is populated. All solutions in $\mathcal{D}_{\tau\succ} \cup \tilde{\mathcal{P}}_\tau$ are finally pooled together and tested for dominance with respect to one another, from where the set $\tilde{\mathcal{P}}_\tau$ is updated accordingly for the next time split. Dominated solutions in the set $\mathcal{D}_{\tau\succ} \cup \tilde{\mathcal{P}}_\tau$ may either be discarded from the solution tracking process or appended to the set $\mathcal{D}_\tau$ for the next time split. The most recently updated set $\tilde{\mathcal{P}}_\tau$ is kept in memory storage and made available to the search algorithms for (optional) use in jump-start optimisation procedures at the beginning of the next time stage. A pseudo-code description of the basic steps of the process is given in Algorithm 9.1 for the case in which infeasible solutions in $\tilde{\mathcal{P}}_\tau$ are discarded from this tracking process but in which newly dominated solutions in $\tilde{\mathcal{P}}_\tau$ are kept. Additionally, a graphical illustration of this procedure may be found in Figure 9.7.

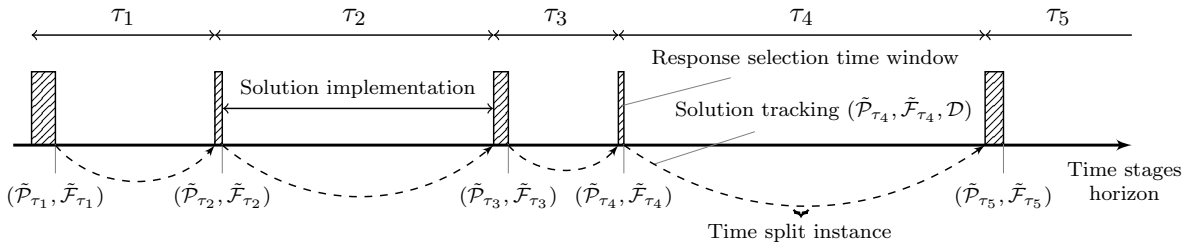


FIGURE 9.7: *The solution tracking process for use in on-line MLE response selection optimisation.*

9.5.2 Evolutionary mechanisms considerations

The tracking algorithm proposed in the previous section exhibits certain similarities to the adaptive evolutionary mechanisms discussed in §2.7.3. In particular, the set $\tilde{\mathcal{P}}_\tau \cup \mathcal{D}_\tau$ may be thought of as a population of individuals to be evolved, while the duration of a time split may be thought of as the duration of a generation, the set $\tilde{\mathcal{P}}_\tau$ may be thought of as the fittest individuals in the population and the discarded solutions may be thought of as the deceased individuals. Crossover between individuals is, however, not applicable in this tracking algorithm, and so the population size monotonically decreases from one generation to the next.

The pseudo-code description in Algorithm 9.1 is, as mentioned earlier, only a special case, and it is duly noted that it is also easily possible to maintain an almost fixed population size of tracked individuals throughout the time stage. In particular, invalid solutions may be transformed into valid ones using a repair operator. Revisiting the chosen implemented solution from the example in §8.6, suppose that approximately thirty five minutes have elapsed since the start of the time stage, such that VOIs 2 has already been visited and serviced by MLE resource f. Then, all time splits initiated past this point in time ought to consider the initially implemented solution as an invalid one. Using a simple string transformation, however, it is possible to convert it back into a new, valid string, by removing VOI 2 from the initial string so that this solution may still be considered in the current population of individuals (except perhaps in the case where a copy of a solution is created). Note, however, that this does not imply that the transformed string will necessarily be a feasible one (although a repair operator may also be employed to address this feasibility issue).

The duration of a time split is not fixed, and will vary based on factors such as the number of solutions tracked at the beginning of the time split (*i.e.* the current population size), the

Algorithm 9.1: Solution tracking algorithm

Input : A set $\tilde{\mathcal{P}}_\tau$ of non-dominated solutions to the MLE response selection problem found during the response selection time window of time stage τ, together with a strategically-chosen set $\mathcal{D}_\tau$

of ancillary solutions **Output**: An evolved, filtered set of non-dominated solutions $\tilde{\mathcal{P}}_\tau$ to the MLE response selection situation in real time

while $\tau = \tau$ **do**
  Take a snapshot of the MLE database for input data updates
  Reevaluate the problem decision and objective spaces
  forall $\boldsymbol{x} \in \tilde{\mathcal{P}}_\tau \cup \mathcal{D}_\tau$ **do**
    Reevaluate solution $\boldsymbol{x}$ in decision and objective space
    if $\boldsymbol{x}$ *is invalid* **then**
      $\tilde{\mathcal{P}}_\tau \leftarrow \tilde{\mathcal{P}}_\tau \backslash \{\boldsymbol{x}\}$
      $\mathcal{D}_\tau \leftarrow \mathcal{D}_\tau \backslash \{\boldsymbol{x}\}$
  forall $\boldsymbol{x} \in \tilde{\mathcal{P}}_\tau$ **do**
    if $\boldsymbol{x}$ *is infeasible* **then**
      $\tilde{\mathcal{P}}_\tau \leftarrow \tilde{\mathcal{P}}_\tau \backslash \{\boldsymbol{x}\}$
  $\mathcal{D}_{\tau\succ} \leftarrow \emptyset$
  forall $\boldsymbol{x} \in \mathcal{D}_\tau$ **do**
    if $\boldsymbol{x}$ *is feasible* **then**
      $\mathcal{D}_{\tau\succ} \leftarrow \mathcal{D}_{\tau\succ} \cup \{\boldsymbol{x}\}$
      $\mathcal{D}_\tau \leftarrow \mathcal{D}_\tau \backslash \{\boldsymbol{x}\}$
  forall $\boldsymbol{x} \in \mathcal{D}_{\tau\succ}$ **do**
    forall $\boldsymbol{y} \in \mathcal{D}_{\tau\succ}$ **do**
      if $\boldsymbol{x} \succ \boldsymbol{y}$ **then**
        $\mathcal{D}_{\tau\succ} \leftarrow \mathcal{D}_{\tau\succ} \backslash \{\boldsymbol{y}\}$
        $\mathcal{D}_\tau \leftarrow \mathcal{D}_\tau \cup \{\boldsymbol{y}\}$
  forall $\boldsymbol{x} \in \tilde{\mathcal{P}}_\tau \cup \mathcal{D}_{\tau\succ}$ **do**
    forall $\boldsymbol{y} \in \tilde{\mathcal{P}}_\tau \cup \mathcal{D}_{\tau\succ}$ **do**
      if $\boldsymbol{x} \succ \boldsymbol{y}$ **then**
        $\tilde{\mathcal{P}}_\tau \leftarrow (\tilde{\mathcal{P}}_\tau \cup \mathcal{D}_{\tau\succ}) \backslash \{\boldsymbol{y}\}$
        $\mathcal{D}_\tau \leftarrow \mathcal{D}_\tau \cup \{\boldsymbol{y}\}$

repartition of solutions tracked in both the sets $\mathcal{D}_\tau$ and $\tilde{\mathcal{P}}_\tau$, as well as the general rate of removal of solutions from the tracking process as a time stage evolves (*i.e.* the death rate). As mentioned earlier, because the time at which disturbances manifest themselves is unknown, so is the time that elapses during the course of a time stage. Unlike most evolutionary procedures, where the size of the initial population may be configured accordingly, and where the population size may even be maintained or increased throughout the evolution process, the solution tracking algorithm faces great uncertainty in respect of effectively assessing ideal initial population conditions.

At this point, one may argue that this issue may be circumvented simply by appending a very large number of solutions uncovered during the course of response selection optimisation processes to the initial set $\mathcal{D}_\tau$ (the set $\tilde{\mathcal{P}}_\tau$ being fixed). Notwithstanding, define the *recoil time* as the amount of time that has elapsed between the end of the last completed time split process and the start of a new time stage. For similar reasons to those mentioned in §9.3 (namely that dynamic input parameters may, meanwhile, have evolved in real time for the duration of this recoil, in turn causing the set $\tilde{\mathcal{P}}_\tau$ possibly to have deteriorated or to no longer be a fully accurate representation of reality), it is easy to see why the initial population size should not be traded off for an inconveniently large expected recoil time. Indeed, on the one hand, setting the initial population size too low might not present an acceptable non-dominated set of solutions in terms of both quantity and diversity at the start of the following time stage, thus rendering the solution tracking component ineffective or inaccurate to some extent. This is an even greater concern whenever a long time elapses between the manifestation of consecutive disturbances and/or the geometry of the feasible decision space is highly sensitive to input parameter deviations. On the other hand, however, setting the initial population too high increases the expected recoil time.

9.6 The optimisation tracker component

Up to this point in the dissertation, the process of searching for good solutions in the decision space of the current MLE response selection problem instance was only assumed to take place at the start of a time stage. Realistically speaking, however, although one may point out that the process of seeking the most preferred alternative during a response selection time window is inseparable from the need to launch search techniques capable of finding the corresponding solution to this alternative, the converse is not true. In other words, searching for good alternatives in this dynamic multiobjective optimisation problem may be conducted independently in time from response selection time windows.

The problem of multiobjective optimisation in a non-stationary environment may be thought of as tracking a series of time-dependent Pareto fronts; the aim of the optimisation tracker component is therefore to continually track the shift in Pareto front approximation at discrete intervals in time, solving the same fundamental mathematical model as the one employed in the model configuration component (with varying dynamic input parameters), by adopting an adequate array of solution search techniques (such as the ones proposed for inclusion in the optimisation methodology component). Unlike the solution tracking component, however, this component operates independently from the MLE response selection DSS clock cycle (*i.e.* independently from the time stages horizon), and on a parallel time line.

The process of optimisation tracking operates as follows. Updates in input data pertaining to MLE response selection operations (again, no matter how small) are first received from external data sources at the start of an optimisation tracking clock cycle, referred to here as a *time slot* $\hat{\tau}$. These time slots are configured to take place at deterministic points in time by bounding the time

dedicated to each problem instance in the form of a sequence of static problem formulations, and so a new time slot is initiated immediately after the previous one has come to an end. Moreover, all updates in input data occurring within a specific time slot are postponed to the beginning of the following time slot[4]. Similarly to the solution tracking process, these dynamic input data are then used to recalibrate all parametric values and set elements in the present mathematical model formulation. Using this updated model, an updated, high-quality set of non-dominated solutions is then sought, where the search termination criterion is analogous to the duration of the time slot. This component's output is then sent and stored in the MLE database for possible use in jump-start optimisation procedures at the start of the following time stage. A graphical illustration of this procedure is portrayed in Figure 9.8.

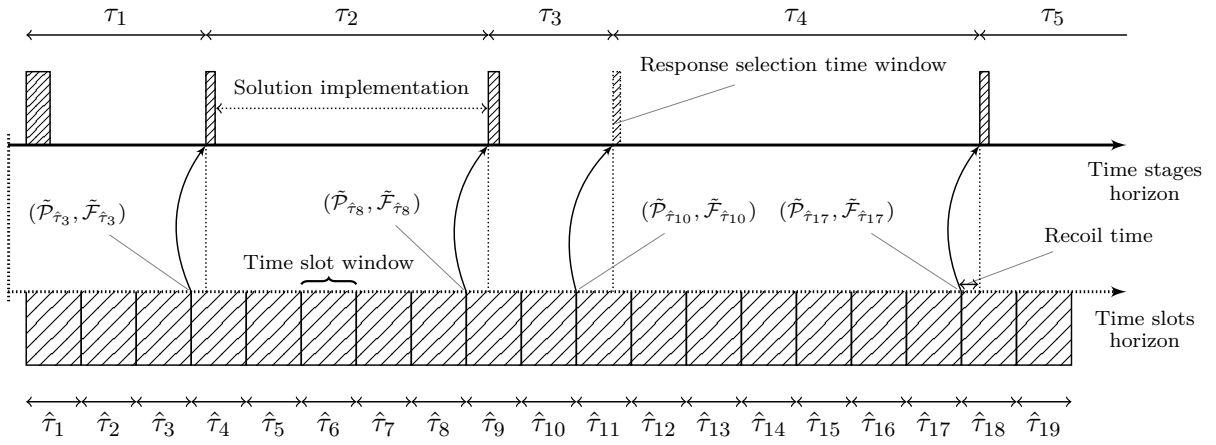


FIGURE 9.8: *The optimisation tracker process for use in on-line MLE response selection optimisation.*

Note that the graphic in Figure 9.8 is not scaled realistically; it is only depicted as such for demonstrative reasons. Moreover, in this simplified illustration, time slots are configured to occur at steady intervals, but it is acknowledged that the length of a time slot should actually be configured based on two independent key factors.

The first factor relates to the amount of variation in input data that has taken place during the previous time slot and, more specifically, to the frequency and types of landscape shifts that have manifested themselves in a consecutive fashion during the course of the previous time slot. The infrequent occurrence of (less perturbing) Category 1 and 2 landscape shifts taking place during the course of a time slot, for example, do not require the next time slot duration to be large.

The second factor is more intricate; it defines a risk trade-off associated with the recoil time which, in this case, refers to the time deficit experienced between the start of a new time stage and the end of the last time slot. Unlike the recoil time period experienced in the solution tracking process, the one found in the optimisation tracker process is much more pronounced. This is because the durations of time slots are expected to generally be much longer than that of time splits. Subsequently, there exists an interesting trade-off between search performance and expected recoil time when considering the appropriate length of time slots in general, where longer time slot configurations allow for a search process to unravel higher-quality non-dominated fronts as a direct result of longer optimisation computational time, but also increase the expected recoil time. Interestingly, it is important to point out that, if disturbances were assumed to occur at deterministic (predictable) points in time, then the recoil time could be nullified by means of

[4]If possible, the latest updates of the decision and objective spaces should alternatively be transferred from the solution tracking component to achieve higher efficiency.

synchronising time slot windows with the advent of disturbances.

It is therefore suggested that time slot durations be determined dynamically as a function of the problem landscape shifts that have taken place recently as well as the (fixed) subjective risk associated with time recoils. Finally, it is suggested that the technique of temporal retraction of foreseen dynamic information (described in §9.3) also be employed as a means to reduce the deterioration of solutions generated during a time slot, particularly in cases where these windows are configured over longer periods of time.

9.7 Fusion scheme for on-line multiobjective optimisation

Based on the observations made in §9.5, it appears that the solution tracker process excels whenever disturbances tend to occur more frequently, provided that a large enough population is evolved every time. This is because non-dominated solutions evolved in an ever-changing environment for relatively shorter periods of time are less likely to deteriorate to a dominated status and/or to become infeasible and/or to become invalid, to the point where a satisfactory Pareto front approximation may be employed for use in jump-start optimisation procedures. On the other hand, it is anticipated that the optimisation tracker process described in §9.6 will be most useful whenever time stages carry on for longer periods of time. This is mainly because optimisation tracking becomes redundant whenever disturbances are triggered more frequently than general time slot duration configurations. Furthermore, it is evident that both components ought to perform better whenever unforeseen variations in input data occur less frequently and/or whenever problem landscape shifts generally perturb the geometry of the current domain and objective spaces to a lesser extent.

9.7.1 Framework description

Instead of only adopting *either* one of the two components mentioned above for implementation in an on-line, dynamic multiobjective optimisation DSS, it is thought that an ideal dynamic information management structure for jump-start optimisation ought to adopt a fusion scheme that incorporates *both* these component. A proposed framework of this type is illustrated in Figure 9.9 and the information sharing process between the components involved in this framework is displayed in Figure 9.10.

Because this consolidated framework consists mostly of material already explained in detail in the previous two sections, the working of the underlying process in this proposed scheme is briefly described as follows. It is suggested that the interaction between the solution tracking component and the response selection time windows be as described in §9.5, with the exception that the solution tracking process now operates on a time continuum instead of having to be reinitialised at the end of every response selection time window. This is because the solution tracking component in this framework also interacts with the optimisation tracker component, independently from the time stages time line. Besides receiving the input sets $\tilde{\mathcal{P}}_\tau$ and $\tilde{\mathcal{F}}_\tau$ from the optimisation methodology component at the end of a response selection time window, together with a set of ancillary solutions $\tilde{\mathcal{D}}_\tau$ to complement these solutions, the solution tracking process in this framework also receives the input sets $\tilde{\mathcal{P}}_{\hat{\tau}}$ and $\tilde{\mathcal{F}}_{\hat{\tau}}$ from the optimisation tracker component at various points in time corresponding to the end points of time slots.

At the end of a response selection time window, the best approximation of the Pareto-optimal set and corresponding Pareto front uncovered during the solution search process of the optimisation methodology component are immediately stored in the MLE database, together with

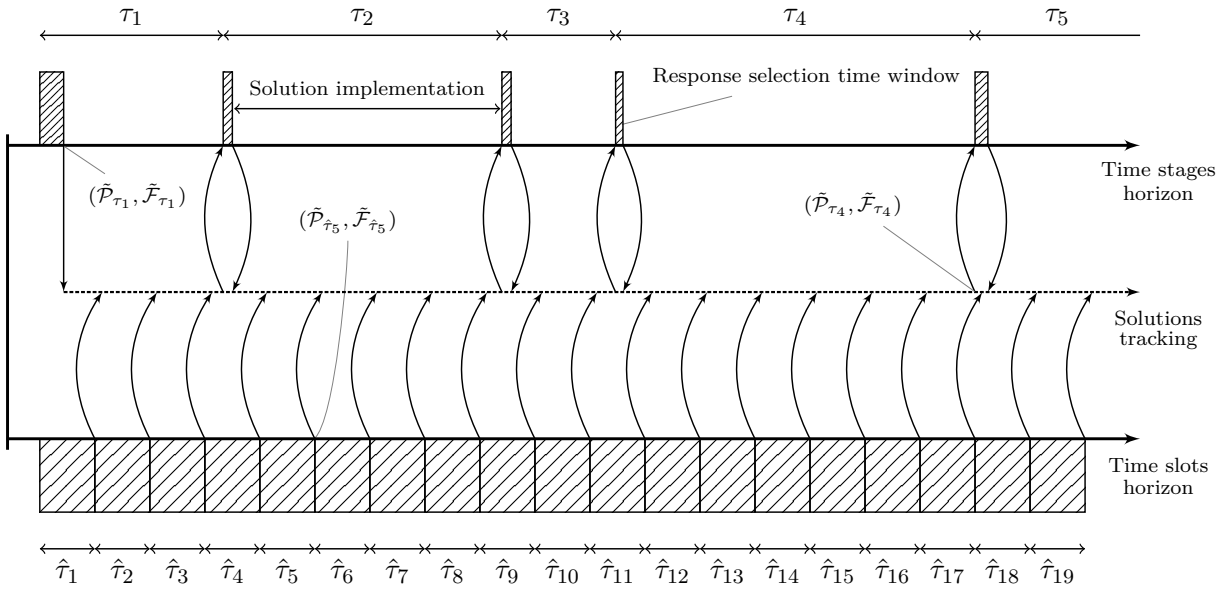


FIGURE 9.9: *Proposed on-line, dynamic multiobjective optimisation fusion scheme.*

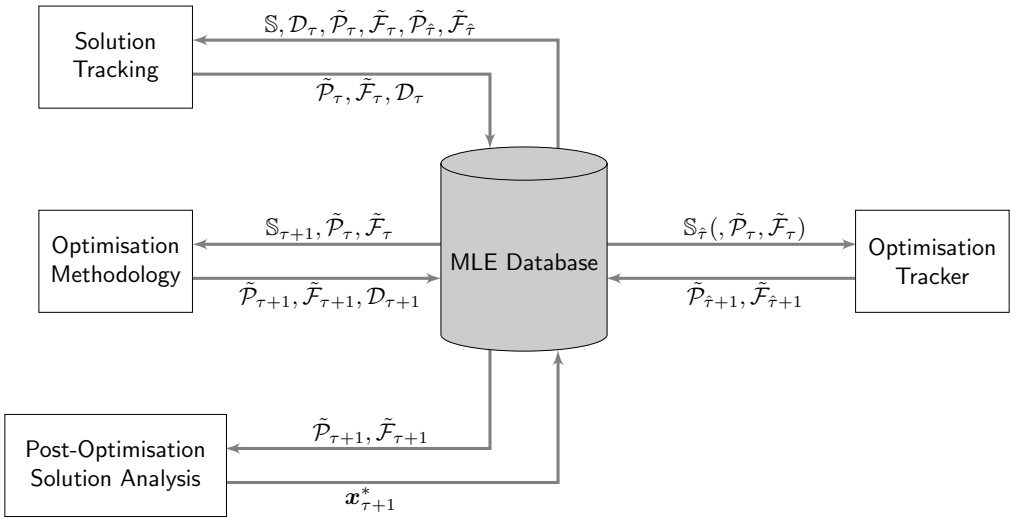


FIGURE 9.10: *Information sharing process amongst the components involved in the fusion scheme of Figure 9.9.*

an adequate set of ancillary solutions. These sets are then presented to the solution tracking component and merged with the current solution tracking population as soon as the next time split is initiated. Meanwhile, high-quality non-dominated sets of solutions and corresponding non-dominated fronts uncovered in a parallel time line in the optimisation tracker component are immediately stored in the MLE database and are also presented to the solution tracking component and merged with the current solution tracking population as soon as the next time split is initiated.

Similarly to the standard solution tracking process of §9.5, these solutions are then evolved in time, and the latest evolved sub-population of elite individuals is updated and stored in the MLE database. As soon as a disturbance is triggered, this most recently found set of non-

dominated solutions and the corresponding non-dominated front are then transferred from the MLE database to the optimisation methodology component for use in jump-start optimisation taking place in the corresponding new response selection time window. Meanwhile, the solution tracking process continues to operate as normal, possibly receiving input originating from optimisation tracking output taking place during the current response selection time window (see, for example, time slot $\hat{\tau}_{17}$ output in Figure 9.9).

9.7.2 Advantages, recommendations and hidden benefits

The presence of this merged framework is imperative as a means to meet the benefits foresaw in employing jump-start optimisation, as the solution tracking and optimisation tracker components might, on their own, not be able to meet the required standards consistently. On the one hand, if any two disturbances occur within a small time interval, then the framework should be able to rely on the sets $\tilde{\mathcal{P}}_\tau$, $\tilde{\mathcal{F}}_\tau$ and $\tilde{\mathcal{D}}_\tau$ (transferred to the solution tracking component at the end of the response selection time window in time stage τ) to provide the response selection time window in time stage $\tau+1$ with a satisfactory estimation of the set of Pareto optimal solutions, as the optimisation tracker may not be afforded enough time to run a complete search during the course of that small time interval. On the other hand, if any two disturbances are triggered far apart, then the more recently evolved sets $\tilde{\mathcal{P}}_{\hat{\tau}}$ and $\tilde{\mathcal{F}}_{\hat{\tau}}$ ensure that the required (Pareto optimality) quality standards are met, as the sets tracked from the previous response selection time window might be significantly reduced and deteriorated in quality as a result of numerous and/or very perturbing problem landscape shifts which may have manifested themselves during the course of the time stage.

Although not depicted in Figure 9.9, yet as hinted at in Figure 9.10, this process may furthermore be improved by making use of jump-start optimisation techniques within the optimisation tracker component itself. That is, by considering the latest, best approximation of the Pareto optimal set of solutions and corresponding Pareto front updated during the solution tracking process as part of the optimisation tracker input data at the start of a time slot.

Unlike the solution tracking algorithm of §9.5, the population of individuals evolved in this framework is not fixed, but rather varies based on the number of solutions generated by the optimisation tracker component[5]. Because these regularly appended sets of solutions are able to provide the population with "elite new-borns" in real time, the potentially tedious task of constructing a set of ancillary solutions at the end of a response selection time window may be significantly relaxed (especially if time slot outputs are made available at a higher rate), or even ignored.

As observed in §9.1, a difficult task pursued in configuring the problem instance reinitialisation component entails establishing threshold metrics capable of measuring significant, unanticipated variations in input data. The configurations of the solution tracking and optimisation tracker processes in this framework are strongly influenced by the general frequency of disturbances, which is itself strongly influenced by the "strictness" of these threshold metrics. Because such configurations may prove to be impractical or too tedious, it is recommended that, in addition to training the problem instance reinitialisation component to detect significant variations in input data, the solution tracking component should also be trained to track the state of the currently implemented solution (which forms part of the solution tracking population) in both decision and objective space, as a means to trigger an alternate class of disturbances. For example, if it is assumed that operator preferences remain the same on the short term, then a new non-dominated

[5]Objective space filtering procedures, such as those presented in §8.6, may alternatively be used as a means to control the size of this output.

solution received from the optimisation tracker component (*i.e.* one that did not form part of the non-dominated set of solutions generated during the last response selection time window) which is mapped in close proximity to the currently implemented solution in objective space might be more desirable from the view point of the operator. More importantly, as a result of small, yet accumulated, changes in problem input data, it is possible for the currently implemented solution eventually to shift into infeasible decision space, or to become dominated in objective space, yet completely unnoticed by the problem instance reinitialisation component. Provided that the necessary HMI features are in place to interact with this system component, another use of the solution tracking process is therefore to allow an MLE response selection operator or decision maker to track the status of the currently implemented solution as it evolves throughout a time stage. In particular, disturbances caused by operator-induced events (see Disturbance 13 in §9.1) may be triggered by designing the solution tracking component to alert the operator of the occurrence of one of the above-mentioned scenarios.

Lastly, and most relevant, it is anticipated that were this on-line optimisation framework to operate effectively, the most recently stored set of non-dominated solutions presented to the solution search subsystem at the start of a time stage might actually be good enough to render the need for further optimisation during the corresponding response selection time window redundant. Significant solution deterioration prevention may, in turn, be achieved if the allocated optmisation time were to be subtracted from the estimated duration of the response selection time window (recalling that processes such as model management and post-optimisation solution analysis still have to be conducted). Alternatively, a highly exploitative jump-start optimisation algorithm may be employed using this *a priori* set of non-dominated solutions over a much shorter computational time than would be required by more traditional (*i.e.* non jump-starting) optimisation procedures, in order to reach a satisfactory standard of approximated Pareto front quality for use during post-optimisation solution analysis.

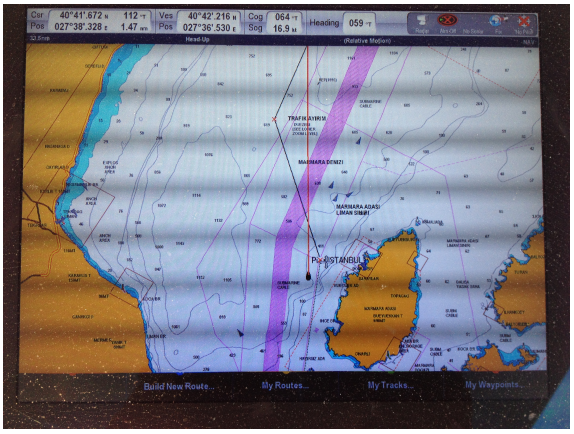
9.8 Micro management of MLE resources

It was motivated in Chapter 6 that a solution selected at the beginning of a time stage and implemented for MLE response selection purposes will very rarely turn out exactly as planned. For this reason, various stochastic model constituents were proposed in an attempt to manage the risk associated with real-time solution deviations from the initial plan of operations. This takes place at a macroscopic decision making level, where all routes are planned simultaneously, and where all solution rectification procedures depend on the problem instance reinitialisation component.

One aspect of MLE response selection that still has to be mentioned pertains to the management of MLE resources on a microscopic decision making level. More specifically, it may be reasonable to assume that MLE resources are operated in real time by a competent crew having the freedom to perform various operational manoeuvres with the assistance of an array of electronic infrastructure including object detection radars, communication with on-land operators and two-way streamed, real-time access to the MLE response selection database. Examples of electronic HMI displays that may be found on board MLE resources are shown in Figure 9.11.

As they bear no macroscopic decision making powers, MLE resources are required to orderly intercept the VOIs assigned to them by the central operator. Although navigation travel arcs forming visitation routes are pre-calculated by the DSS, these are mostly included to provide general heading instructions for each MLE resource at any time, provided that it maintains a fixed average travelling speed. Due to the stochastic elements of this problem, however, MLE

(a) *Technical information pertaining to an MLE resource.*



(b) *Radar information for detecting and tracking objects at sea.*

FIGURE 9.11: *Examples of HMI displays that may be found on board MLE resources.*

resource course and speed adjustments are bound to be made throughout solution implementation operations. In addition, and as discussed in §6.5, not only should the servicing of VOIs be conducted effectively, but also as quickly as possible. These operational manoeuvres ultimately ought to be conducted in a way that agrees with the MLE response selection goals pursued by the coastal nation. In particular, although efficient microscopic decision making[6] of this kind will not affect the visitation score or consensus level, it may very well reduce operating costs and total delay time, thus pursuing Objectives II and III of Chapter 4.

9.9 Chapter summary

In the first section of this chapter, the problem instance reinitialisation component of the MLE response selection DSS put forward in this dissertation was described in detail using various examples of disturbances and reinitialisation protocols. A theoretical discussion pertaining to the shift in decision and objective space of a problem instance as a result of input data variations was then conducted in §9.2, and this was followed in §9.3 by a description of the technique of temporal retraction of foreseen dynamic input parameters, and an analysis of the jump-start optimisation prerequisites required for solving the MLE response selection problem in an on-line fashion in §9.4. Full descriptions of the solution tracking and optimisation tracker components of the proposed DSS were provided in §9.5 and §9.6, respectively, after which an on-line, dynamic optimisation framework for fusing the results of these two components together was proposed in §9.7. Finally, the role of MLE resources in making effective micro-decisions in agreement with the goals of the coastal nation was explained in §9.8.

[6] Auxiliary objectives of this kind are often referred to as *mean* objectives in the literature [27].

CHAPTER 10

Conclusion

Contents

The dissertation closes in this chapter with a summary of the work contained therein, as well as a recapitulation on effective ways in which to manage the various intricacies and complications associated with situation redeployment in the context of the MLE response selection problem. An appraisal of the contributions of this dissertation and a discussion on possibilities for related future work are also provided in closing.

10.1 Dissertation summary

The dissertation opened in Chapter 1 with a review of certain international laws pertaining to activities at sea in order to establish a context for the complex MLE challenges faced by coastal nations. An informal description of the MLE response selection problem was presented, illustrated and supplemented with motivating arguments underlying the need for designing a DSS that may be employed by MLE response selection operators when solving instances of the problem. This was followed by a discussion on the aim and scope of the dissertation, as well as the objectives to be pursued. The chapter closed with a description of the organisation of material in the dissertation.

The various underlying concepts presented in the dissertation with respect to the design of an MLE response selection and routing DSS were introduced in Chapter 2, in fulfilment of Dissertation Objective I of §1.4. In §2.1, a discussion was conducted on the philosophy and fundamental concepts behind decision making, and this was followed by an introduction to decision making problems involving multiple conflicting objectives. A mathematical description of the notion of dominance and Pareto-optimality was next provided and the section closed with a discussion on multiperson decision making. In §2.2, a general discussion on metaheuristic and other solution techniques for optimisation problems was conducted, and a method of classifying these techniques was proposed. The popular use of metaheuristics for solving complex multiobjective optimisation problems in the literature was also motivated in this section. Following this, the nature and working of the method simulated annealing and of evolutionary algorithms were

outlined and discussed thoroughly in §2.3 and §2.4, respectively, both within the context of single and multiobjective optimisation problems. In addition, the methods of tabu search, ant colony optimisation and particle swarm optimisation were described in §2.5 within the context of single objective optimisation problems. A brief review of the classical VRP and a number of its variants was then presented in §2.6, with an emphasis on the role of the nature of information with respect to defining, modelling and solving difficult VRP variants, as well as on the principles behind stochastic and dynamic VRPs. The focus next turned in §2.7 to a discussion on various approaches adopted in the literature for solving dynamic multiobjective optimisation problems. Finally, a review of MLE DSSs in existence was conducted in §2.8, with an emphasis on MLE resource assignment DSS. Two high-quality relatable studies were also summarised in some detail in this final section of the chapter.

The design of a novel, generic system architecture for an MLE DSS was proposed in the first section of Chapter 3, with particular emphasis on the subsystems forming the MLE response selection DSS, in fulfilment of Dissertation Objective II of §1.4. The flow of events between the threat detection, threat evaluation and MLE response selection subcycles was then highlighted in §3.2. Finally, three generic decision making paradigms were presented in §3.3, capturing the roles of the various decision making entities when solving the MLE response selection problem with the aid of the proposed DSS architecture, in fulfilment of Dissertation Objective III of §1.4.

In Chapter 4, the process of selecting, populating and storing the fundamental mathematical structures in the mathematical modelling subsystem was described, in fulfilment of Dissertation Objective IV of §1.4. The chapter opened with a discussion on the underlying assumptions and concepts which are critical to modelling the MLE response selection problem effectively. Then, in the following three sections, fundamental components for modelling the MLE response selection problem in a centralised, intermediate or decentralised decision making paradigm were presented in detail.

A selection of important dynamic features and constraints that may be incorporated in the mathematical modelling process so as to accommodate a variety of special requests or instructions by operators, were explored in Chapter 5. These features form part of the so-called model management component of the proposed MLE response selection DSS. The two subcomponents of the model management component, which is responsible for the inclusion of these constraints, were first introduced in §5.1. These subcomponents are the cutting plane subcomponent, responsible for directing the solution search process in suitable directions (by incorporating operator expertise as part of the input data), and the model adaptation subcomponent, in which situations that are not accounted for in the generic model configuration component may be incorporated in the formulation for a particular problem instance. The first subcomponent above was described in some detail in §5.2, while the second subcomponent was subsequently described in more detail in §5.3. A number of general comments were finally made in §5.4 in respect of incorporating the model management component in a real-life MLE response selection DSS.

In fulfilment of Dissertation Objective V of §1.4, an array of stochastic components that may be incorporated into the mathematical modelling process as a means to allow operators to manage the risk of uncertainty associated with the various probabilistic elements of the MLE response selection problem, were designed in Chapter 6. The MLE response selection problem was first described in some detail within a stochastic paradigm in §6.1, and this was followed in §6.2 by an investigation on effective ways in which to model the uncertainty pertaining to the positions of VOIs in space. A step-by-step construction of model components that incorporate the uncertainty associated with VOI visitation locations was put forward §6.3, and §6.4 contained a methodology for accommodating stochasticity of route travel distances. In §6.5, a brief discussion on the uncertainty pertaining to VOI service times was conducted, after which the

management of uncertainty pertaining to the threatening nature of VOIs was addressed in §6.6. Finally, the focus shifted in §6.7 from the impact that stochastic elements have on the feasible decision space to the impact they have on the evaluation of solution performances in objective space.

An algebraic description was derived for approximating interception points between MLE resources and VOIs within a simplified planar context in the opening section of Chapter 7. A method was next proposed in §7.2 for performing end-of-route assignments to MLE vessels incorporating a sub-process of the optimisation methodology. This was followed by a suggested generic approach towards encoding solution strings as well as performing overall solution transformations in §7.3, and a discussion on Pareto front approximation in §7.4. Five solution search techniques capable of solving instances of the MLE response selection problem approximately were then put forward, in fulfilment of Dissertation Objective VI of §1.4. A dominance-based archived multiobjective simulated annealing approach towards solving the MLE response selection problem was discussed in depth in §7.5. In the next section, the NSGA-II was described as an alternative solution process for solving the MLE response selection problem. An idea for a novel hybrid metaheuristic, combining the population based, non-dominated sorting procedure and diversity preservation rule of the NSGA-II with the Metropolis rule of acceptance and archiving mechanism of the multiobjective simulated annealing method, was then proposed in §7.7. Finally, two multistart archived multiobjective simulated annealing methodologies were adapted for use in a parallel computation paradigm in §7.8.

In Chapter 8, a subset of the optimisation methodologies proposed in Chapter 7 were tested in respect of their abilities to yield high-quality solutions to two hypothetical instances of the MLE response selection problem, in fulfilment of Dissertation Objective VII of §1.4. Following a description of the performance measure employed for assessing the quality of non-dominated fronts in §8.1 (with particular focus on the hypervolume quality indicator), an estimation of the domain sizes of MLE response selection problems followed in §8.2. A lower-complexity hypothetical MLE response selection scenario, together with a representative mathematical model, were then put forward in §8.3, and a number of experiments were performed in §8.4, where the results of these experiments were reported and analysed in detail. These experiments were next repeated in §8.5 in the context of a higher-complexity hypothetical scenario using an appropriate selection of solution search techniques. Finally, a description of the post-optimisation component of the MLE response selection DSS proposed in this dissertation was provided in §8.6, using various hypothetical graphical illustrations.

The dynamic features of the MLE response selection problem were finally analysed in greater detail in Chapter 9, with the aim of consolidating the evolution of information into a multi-level framework capable of triggering rapid and effective problem instance reinitialisation procedures, in fulfilment of Dissertation Objective VIII of §1.4. The problem instance reinitialisation component was described in detail, referring to various examples of disturbances and reinitialisation protocols in the first section of the chapter, following which a theoretical discussion pertaining to the nature of the shift in decision and objective space of a problem instance as a result of input data variations was conducted in §9.2. A description of the technique of temporal retraction of foreseen dynamic input parameters was next provided in §9.3, together with an analysis of the jump-start optimisation prerequisites required for solving the MLE response selection problem in an on-line fashion in §9.4. The solution tracking and optimisation tracker components were then described in detail in §9.5 and §9.6, respectively, after which a framework for solving on-line, dynamic multiobjective optimisation problems in general was proposed in §9.7. Lastly, the importance of the role of MLE resources in making effective micro-decisions that agree with the goals of the coastal nation was motivated in §9.8.

10.2 Overcoming situation redeployment challenges

Throughout the dissertation and, in particular, as a result of the observations made in Chapter 8, it was acknowledged that the process of effectively resolving the MLE response selection problem by means of the proposed DSS faces several challenges in respect of rapidly identifying an adequate solution to a particular situation. These (not necessarily independent) challenges were found to be primarily caused by: (a) very large and often fragmented and/or disjoint problem domain spaces, (b) limited response selection solution computational time, (c) large Pareto fronts in objective space (as a result of employing four or more objective functions), (d) solution deterioration during response selection time windows (as a result of the instance snapshot effect), (e) unpredictable problem reinitialisation times and (f) relatively high uncertainty surrounding the progression and outcome of an implemented solution.

An array of (again, not necessarily independent) ideas and concepts were consequently suggested and developed throughout the dissertation as a means to restrain the impact caused by the various challenges mentioned above. To recapitulate, these challenges may be addressed to a certain extent by incorporating a subset of the following concepts in the DSS proposed in this dissertation:

1. Implementing a decentralised decision making paradigm as a means to partition and reduce the domain space into multiple independent sub-problems (see §3.3 and §4.4), partially addressing the challenge in (a) above,

2. Utilising the *a priori* expertise of human decision makers as a means to filter away unwanted portions of the domain space (see Chapter 5), partially addressing the challenge in (a) above,

3. Adopting lower complexity computing languages to accelerate the execution speed of the solution search methodologies, addressing to some extent the challenge in (b) above,

4. Developing, implementing and identifying solution search techniques that thrive in high-complexity environments (see Chapters 7 and 8), partially addressing the challenges in (b) and (c) above,

5. Making use of parallel computing in the form of multiple processing and memory units (see §7.8), partially addressing the challenges in (b) and (c) above,

6. Employing intelligent solution transformation procedures (see Chapter 7), contributing toward addressing the challenges in (a) and (b) above,

7. Applying the temporal retraction of foreseen dynamic input parameters technique at the start of response selection time windows (see §9.3), addressing the challenge in (d) above,

8. Educing an array of chance constraints and other stochastic programming model constructs (see Chapter 6), addressing the challenge in (f) above,

9. Designing an on-line, dynamic multiobjective framework capable of facilitating response selection instance reinitialisation procedures at any time (see §9.7), partially addressing the challenges in (b), (c) and (e) above, and

10. Training MLE resource crews in making rational and effective micro-management decisions during the course of MLE response selection operations (see §9.8), contributing toward addressing the challenge in (f) above.

10.3 Dissertation contributions

The main contributions of this dissertation are fivefold, as outlined in this section.

Following the review in Chapter 2 of MLE DSSs in existence, it was acknowledged that very few studies tend to focus on the MLE response selection process for active MLE resources in real time, and no previous studies were found in the literature for solving the MLE response selection problem as a combinatorial optimisation problem using the analogy of the dynamic VRP as a means to model the problem. In addition, most of the existing studies were conducted within the context of the environment of a specific coastal nation and thus tend to not be formulated in a generic manner. The first contribution of this dissertation therefore centres around the design of a novel, generic MLE response selection DSS and overall MLE system architecture, as proposed in Chapter 3. In this design, it was acknowledged that the MLE response selection operations of a coastal nation are typically not conducted *via* a single central operator (or group of central operators) assisted by a single DSS, but are rather orchestrated by multiple, and often to some degree independent, role players. Consequently, three suitable paradigms of decision entity autonomy, inspired by the notion of the *processing* of information in VRPs (as described in §2.6) were proposed in which the MLE response selection framework may function in a practical manner.

The second contribution was the combinatorial optimisation model formulation of a new kind of dynamic multiobjective VRP (defined in the dissertation as the MLE response selection routing problem) in Chapter 4, in which the depots represent the bases from whence MLE resources are dispatched, the fleet of vehicles represents the fleet of MLE resources at the collective disposal of the decision entities and the customers represent the VOIs tracked at sea within the territorial waters of the coastal nation. The formulation of the MLE response selection problem therefore incorporates a unique feature combination of a heterogeneous fleet of vehicles, multiple depots, customer profits, asymmetric travel arc lengths and dynamic customer locations. In addition, the nature of information specification in this problem is classified as both *dynamic* (*i.e.* certain input data are made known to the operator/decision maker in a continual fashion) and *stochastic* (*i.e.* certain input data are estimated or prescribed as probability distributions). Various deterministic fixed and dynamic features of the MLE response selection problem were identified in Chapters 4 and 5, and incorporated into the mathematical modelling subsystem of the DSS. A selection of the fundamental modelling features have been published in [32], while a selection of the dynamic modelling features were published in [33]. Additionally, various stochastic features of the MLE response selection problem were derived in Chapter 6, and also incorporated into the mathematical modelling subsystem of the DSS.

The third contribution of this dissertation involves the incorporation of an array of various intricate features associated with the nature of the MLE response selection problem into the optimisation methodology component, as described in Chapter 7. In particular, the following solution search aspects had to be designed by hand:

1. An algebraic description of a methodology for approximating interception points between MLE resources and VOIs when subject to trajectory deviations caused by environmental vectors.

2. A generic, practical representation of solutions to MLE response selection problem instances, as well as a general process for transforming solutions during a search process which accommodates both fixed and dynamic modelling features (this work has also been published in [33]).

3. Three processes for perturbing solutions from one state to another as part of the method of simulated annealing.
4. An investigation into the careful consideration of subspecies distribution among the initial population of a genetic algorithm, as well as a novel crossover operator procedure that permits inter-species reproduction.
5. A low-complexity algorithm for assigning vehicles to end-of-route vertices during the search process, which functions based on certain model management input data and autonomy metrics associated with active MLE resources.
6. A hybridisation algorithm for solving instances of the MLE response selection problem in which the population-based approach of the NSGA-II is combined with the Metropolis acceptance rule of the method of simulated annealing.
7. Two multistart optimisation search techniques inspired by the archived multiobjective simulated annealing algorithm, for use in a parallel computing paradigm.

The literature on jump-start optimisation techniques for use in on-line, dynamic multiobjective optimisation problems was found to be sparse and/or too problem-specific. The fourth contribution of this dissertation therefore evolves around the bundle of novel concepts introduced in Chapter 9 which, when fused together, forms the basis of a design of an effective, generic, on-line multiobjective framework for use in solving similar optimisation problems to the one studied in this dissertation.

The final contribution relates to the locally pioneering nature of the work pursued in this dissertation, which is believed to open the door to a number of enticing research opportunities and suggestions for future work in both MLE response selection as well as in certain other functional elements of the MLE system. Examples of these future work opportunities are suggested in the following section.

10.4 Future work

In this section, two open questions related to the MLE response selection problem are posed. Six suggestions are also made with respect to possible future research emanating from the work presented in this dissertation.

Question 10.4.1 *Provided that a (hypothetical) coastal nation has adopted a particular MLE response selection decision making paradigm in the past, what impact would a paradigm shift have on overall MLE response selection performance?*

In Chapter 3, basic MLE response selection system architectures were proposed for use in different decision making paradigms. These paradigms relate to the processing of information in respect of the steps followed once new information are made available to the MLE response selection subsystem. The two extreme cases are the centralised decision making paradigm (in which all information is collected and processed by a central DSS operator or group of operators) and the decentralised decision making paradigm (in which a large portion of the information is processed by decentralised DSSs and operators). Model formulations and suggestions in respect of each of these decision making paradigms were provided in Chapter 4.

In this dissertation, an intermediate decision making paradigm was also employed in the hypothetical scenarios of Chapter 8 as a means of validating the system proposed. Although the

results thus obtained are believed to be of satisfactory quality (subject to specific parametric configurations and limited computational time frames), it is unclear how the implementation of such a paradigm would perform when subject to even higher complexity scenarios. More importantly, the performances of the centralised and decentralised decision making paradigms should also be investigated in the context of an array of other hypothetical scenarios.

The method of implementation of a particular decision making paradigm is believed to be very important with respect to the performances of the optimisation methodology employed for solving instances of the MLE response selection problem. Although implementation of the centralised decision making paradigm may seem more beneficial for solving relatively low to medium complexity instances of the MLE response selection problem, it may be the case that certain MLE response selection situations are so complex that even the best central DSS and operator may not be able to generate non-dominated fronts of significant quality within a limited time frame. The partitioning or decomposition schemes that have been proposed in the literature for solving highly complex VRPs should therefore be investigated in the context of the implementation of the decentralised decision making paradigm. It is believed that such schemes may be effective when solving highly complex MLE response selection problem instances, given a relatively small computational time budget.

Question 10.4.2 *Are there alternative paradigms for modelling the MLE response selection problem effectively?*

The MLE response selection problem formulation presented in this dissertation was modelled as a combinatorial optimisation problem inspired by a combination of variants of the capacitated VRP. Although it has been shown to be an effective way of modelling the MLE response selection problem, alternative modelling techniques (such as using, for example, self-organising agents) should also be investigated as a means of populating the model configuration component with additional features. More importantly, suitable ways of combining these alternative techniques in the currently implemented VRP formulation should be investigated as a means of representing the problem more realistically.

Suggestion 10.4.3 *Design an effective, user-friendly HMI for use by MLE response selection operators.*

In the proposed system architecture of Chapter 3, input data are presented to an MLE response selection operator by means of an HMI, allowing him to review certain information as part of his decision making process, such as notifications of disturbances, threat analysis results associated with detected VOIs or non-dominated solutions generated for a particular problem instance. The operator is also required to transmit information to the MLE response selection DSS and other operators *via* this HMI, such as model management configurations (as described in Chapter 5) or post-optimisation solution analysis input values (as described in §8.6).

In order to minimise the time incurred between the detection of a disturbance and the implementation of a (new) time stage solution (*i.e.* the response selection time), it is necessary *inter alia* to design a user-friendly HMI that is able to display comprehensive information on demand as well as allow the operator to easily interact with it in a relatively stress-free manner and within minimal time.

Suggestion 10.4.4 *Design an idle MLE resources DSS.*

As discussed in Chapter 3, the aim of an idle MLE resources management DSS is to deal with the

allocation of MLE resources in an idle state to bases or patrol circuits. The functionality of this DSS is expected to evolve around the expertise and experience of human operators combined with large amounts of historical data (for example, consider the probability distributions of specific threatening activities occurring in specific zones of the jurisdiction area over specific time intervals) and various mathematical models aimed at scheduling idle MLE resources in both time and space in the most effective manner *a priori*. In other words, because the occurrence of newly detected VOIs is stochastically distributed in space and time, it is crucial to manage idle MLE resources in such a way that leaves them, *a posteriori*, in strategic positions of relative readiness so that they may be dispatched rapidly for MLE response selection operations.

Unlike the MLE response selection DSS, which receives input data from the threat detection and threat evaluation subsystems for immediate use in the decision making search process, the idle MLE resources DSS is required to incorporate a subsystem responsible for accumulating historical information derived from observations made in respect of VOIs that have been intercepted, such as the threat nature, location and time at which these VOIs are detected. In addition, another subsystem should be responsible for conveying information (in real time) to the MLE response selection DSS with respect to end-of-route assignment preferences, as described in Chapter 5.

Suggestion 10.4.5 *Design an MLE threat evaluation DSS.*

Following the VOI detection process in an MLE environment, a threat evaluation process should provide the operator with expectations as to the nature of VOIs at sea, the degree of "oddness" associated with VOI behaviour as well as expectations with respect to the levels of threats that these VOIs may pose. This critical information on the threatening nature of VOIs should then be employed as part of the input data to the MLE response selection decision making process considered in this dissertation.

As discussed in the proposed MLE system architecture of Chapter 3, the purpose of the VOI threat analysis DSS is to assist a threat evaluation operator in quantifying the potentially threatening nature of VOIs, by providing automated decision support for analysing the behaviour of VOIs based on the information collected during the threat detection process and by making automated inferences with respect to the natures and levels of threat posed by VOIs. This process therefore only involves system tracks of VOIs and not the tracks of maritime objects deemed of little or no interest. Ultimately, the output of such a DSS should consist of vectors associated with each VOI which contain entries corresponding to estimated probabilities with which the VOI in question belongs to a finite number of known threat classes, an unknown threat class and a false alarm class. Hypothesis testing, based on historical kinematic and other data, is an example of a technique that may be used for this purpose.

Suggestion 10.4.6 *Design an MLE response selection DSS concept demonstrator.*

In order to demonstrate what may be achieved if the proposed MLE response selection DSS were to be developed and implemented in a computing environment accommodating live data streaming, the establishment of a computerised concept demonstrator of this DSS should be considered. Such a demonstrator should include an effective interaction interface for communicating with a (hypothetical) operator in such a way that he does not need to be familiar with the technicalities of the inner working of the DSS in order to be able to use it. In addition, it is recommended that this demonstrator should also include a data and response selection visualisation component capable of playing through an MLE scenario.

Suggestion 10.4.7 *Test the proposed on-line, dynamic multiobjective DSS framework proposed in this dissertation practically.*

Implementing and testing the dynamic optimisation framework of §9.7 in a hypothetical or real-life environment may be perceived as a tedious enterprise, which it most presumably is. Fortunately, because of its generic nature, this framework need not necessarily be implemented in the field of MLE response selection, but may instead be tested in the context of other practical on-line dynamic multiobjective problems, including simplified cases such as dynamic problems in which reinitialisation instance times are known in advance.

Interesting challenges anticipated when testing this framework may include selecting adequate search techniques to be employed for the response selection and time slot windows (independently), as well as adequately configuring the sets of ancillary solutions to be tracked alongside the non-dominated front of a given problem instance during the solution tracking process. For instance, it is suspected that optimisation tracking ought to employ exploratory search techniques as a means to fully access all domain subspaces of the ever-changing problem landscape, while response selection windows ought to apply highly-exploitative search methods to the *a priori* provided set of non-dominated solutions (provided that this set represents a satisfactory approximation of the true Pareto-optimal set of solutions). Moreover, the automated algorithm parametric configurations and search termination criteria in optimisation tracking procedures at the start of a given time slot window will preferably have to be adapted to the types and intensities of problem landscape shifts that have manifested themselves over the course of the preceding time slot window.

Suggestion 10.4.8 *Design efficient multiobjective jump-start optimisation search techniques for solving the on-line, dynamic multiobjective DSS framework proposed in this dissertation.*

In line with Suggestion 10.4.7 above, an important aspect that was omitted from consideration in this dissertation involves developing adequate multiobjective jump-start optimisation techniques in which *a priori* information about the decision and objective spaces of a problem instance may be used to resolve a situation with higher efficiency. More specifically, it is required to adapt standard (non jump-starting) multiobjective search techniques intelligently so as to incorporate certain *a priori* information on the nature of the Pareto-optimal set of solutions and corresponding Pareto front into their search processes. These adapted search techniques may then be tested against their standard form in the context of hypothetical or real-life scenarios by, for example, observing the time difference in front convergence (according to some pre-fixed convergence threshold), or simply measuring the improvement in hypervolume measure achieved over a relatively small, fixed computational time period[1].

While this information inclusion process may seem obvious in the context of certain search techniques, it may be slightly more intricate with others. The NSGA-II, for instance, may simply incorporate a subset of these "elite" individuals as part of its initial population while, if needs be, filling the rest of the population with randomly generated individuals to meet the required initial population size. Moreover, a *jump-start archived multiobjective simulated annealing* may simply be initiated by filling the initial archive with a subset of non-dominated solutions (in contrast to initiating the algorithm with an empty archive). In this adapted algorithm version, however, careful consideration must be given to the configuration of the cooling schedule. More specifically, depending on the quality and quantity of non-dominated solutions populating the

[1]Noting that the aim of pursuing jump-start optimisation is to significantly reduce the computational time required to compute a non-dominated front that is adequate for use in post-optimisation solution analysis procedures.

initial archive, the initial temperature and/or the cooling parameter ought to be set significantly lower. This is because the traditional simulated annealing anterior stages of lenient search space exploration may become redundant.

References

[1] Archetti C, Speranza M & Vigo D, 2012, *Vehicle routing problems with profits*, (Unpublished) Technical Report, Department of Economics and Management, University of Brescia, Brescia.

[2] ASTUTE, 2014, *United Nations Convention on the Law of the Sea*, [Online], [Cited December 19th, 2014], Available from `http://www-dsp.elet.polimi.it/ispg/ASTUTE/wiki/doku.php?id=d22:information_retrieval_and_fusion:multimodal_information_fusion`.

[3] Bäck T, 1994, *Selective pressure in evolutionary algorithms: A characterization of selection mechanisms*, Proceedings of the First IEEE Conference on Computational Intelligence, pp. 57–62.

[4] Banos R, Ortega J, Gil C, Fernandez A & De Toro F, 2013, *A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows*, Expert Systems with Applications, **40(5)**, pp. 1696–1707.

[5] Barán B & Schaerer M, 2003, *A multiobjective ant colony system for vehicle routing problem with time windows*, Proceedings of the 2003 International Conference on Applied Informatics, pp. 97–102.

[6] Bashir F, Khokhar A & Schonfeld D, 2007, *Object trajectory-based activity classification and recognition using hidden Markov models*, IEEE Transactions on Image Processing, **16(7)**, pp. 1912–1919.

[7] Blum C & Roli A, 2003, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*, Journal of ACM Computing Surveys, **35(3)**, pp. 268–308.

[8] Borgerson SG, 2009, *The national interest and the Law of the Sea*, Council on Foreign Relations Press No. 46, Washington (DC).

[9] Bouejla A, Chaze X, Guarnieri F & Napoli A, 2014, *A Bayesian network to manage risks of maritime piracy against offshore oil fields*, Journal of Safety Science, **68**, pp. 222–230.

[10] Bowden A, 2010, *The economic costs of maritime piracy*, (Unpublished) Technical Report, One Earth Future Foundation, Louisville (CO).

[11] Box GE, 1979, *Robustness in the strategy of scientific model building*, Journal of Robustness in Statistics, **1**, pp. 201–236.

[12] Branke J, 1999, *Memory enhanced evolutionary algorithms for changing optimization problems*, Congress on Evolutionary Computation, **3**, pp. 1875–1882.

[13] Branke J, Kaussler T, Smidt C & Schmeck H, 2000, *A multi-population approach to dynamic optimization problems*, Springer, London.

[14] Bressen T, 2007, *Consensus decision making: What, why, how*, Berrett-Koehler Publishers, San Francisco (CA).

[15] Bromiley P, 2003, *Products and convolutions of gaussian probability density functions*, Tina-Vision Memo, **3**, pp. 2–7.

[16] Burger AP, 2014, University of Stellenbosch, [Personal Communication], Contactable at `apburger@sun.ac.za`.

[17] Burke EK & Kendall G, 2005, *Search methodologies*, Springer Science & Business Media Incorporated, New York (NY).

[18] Busetti F, 2003, *Simulated annealing overview*, [Online], [Cited June 2nd, 2013], Available from `http://www.geocities.%20com/francorbusetti/saweb`.

[19] Bynum WF, 2009, *The origin of species by means of natural selection: Or, the preservation of favored races in the struggle for life*, AL Burt, New York (NY).

[20] Cámara M, Ortega J & De Toro F, 2010, *Approaching dynamic multi-objective optimization problems by using parallel evolutionary algorithms*, Springer, Berlin.

[21] Cámara M, Ortega J & De Toro F, 2007, *Parallel processing for multi-objective optimization in dynamic environments*, Proceedings of the IEEE International Symposium on Parallel and Distributed Processing, pp. 1–8.

[22] Cape Town Green Map, 2013, *City takes its marine and environmental law enforcement unit to the next level*, [Online], [Cited January 26th, 2013], Available from `http://www.capetowngreenmap.co.za/blog/city-takes-its-marine-and-environmental-law-enforcement-unit-next-level`.

[23] Castro JP, Landa-Silva D & Pérez JAM, 2009, *Exploring feasible and infeasible regions in the vehicle routing problem with time windows using a multi-objective particle swarm optimization approach*, Springer, Berlin.

[24] Chakraborty N, 2004, *Genetic algorithms in materials design and processing*, International Materials Reviews, **3**, pp. 246–260.

[25] Chen P & Xu X, 2008, *A hybrid algorithm for multi-depot vehicle routing problem*, Proceedings of the 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 2031–2034.

[26] Clarke G & Wright JW, 1964, *Scheduling of vehicles from a central depot to a number of delivery points*, Operations Research, **12(4)**, pp. 568–581.

[27] Clemen R & Reilly T, 2001, *Making hard decisions with decision tools*, Duxbury, Belmont (CA).

[28] Cobb HG & Grefenstette JJ, 1993, *Genetic algorithms for tracking changing environments*, (Unpublished) Technical Report, Naval Research Lab, Washington (DC).

[29] Coello CA, Lamont GB & van Veldhuizen DA, 2007, *Evolutionary algorithms for solving multi-objective problems, 2nd Edition*, Springer Science & Business Media, New York (NY).

[30] Collins English Dictionary, 2013, *Decide*, [Online], [Cited September 4th, 2013], Available from `http://dictionary.reference.com/browse/decide`.

[31] Collins english dictionary, 2014, *Metaheuristic*, [Online], [Cited September 23rd, 2014], Available from `http://dictionary.reference.com/browse/metaheuristic`.

[32] Colmant A and van Vuuren JH, 2013, *Prerequisites for the design of a maritime law enforcement resource assignment decision support system*, Proceedings of the 42nd Annual Conference of the Operations Research Society of South Africa, pp. 90–101.

[33] Colmant A and van Vuuren JH, 2014, *Solution representation for a maritime law enforcement response selection problem*, Proceedings of the 43^{rd} Annual Conference of the Operations Research Society of South Africa, pp. 79–87.

[34] Colmant A, 2012, *A simulation study of the impact of lead time on the costs of deliveries*, BComm Honnours Project, University of Stellenbosch, Stellenbosch.

[35] Czyżżak P & Jaszkiewicz A, 1998, *Pareto simulated annealing: A metaheuristic technique for multiple-objective combinatorial optimization*, Journal of Multi-Criteria Decision Analysis, **7(1)**, pp. 34–47.

[36] Dabrowski JJ & De Villiers JP, 2014, *Maritime piracy situation modelling with dynamic Bayesian networks*, Journal of Information Fusion, **23**, pp. 116–130.

[37] Dantzig GB & Ramser JH, 1959, *The truck dispatching problem*, Management Science, **6(1)**, pp. 80–91.

[38] Darby-Dowman K, Fink RK, Mitra G & Smith JW, 1995, *An intelligent system for US Coast Guard cutter scheduling*, European Journal of Operational Research, **87**, pp. 574–585.

[39] De Seligny J & Grainger R, 2010, *The state of world fisheries and aquaculture*, Food and Agriculture Organization of the United Nations, Rome.

[40] Deb K & Goel T, 1999, *Evolutionary multi-criterion optimization*, Evolutionary Algorithms in Engineering and Computer Science, **1**, pp. 135–162.

[41] Deb K, Pratap A, Agarwal S & Meyarivan T, 2002, *A fast and elitist multiobjective genetic algorithm: NSGA II*, Evolutionary Computation on IEEE Transactions, **6(2)**, pp. 182–197.

[42] Deb K, 2001, *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons, Beijing.

[43] Dembczyński K, Kotłowski W & Słowiński R, 2006, *Additive preference model with piecewise linear components resulting from dominance-based rough set approximations*, Springer, Berlin.

[44] Deshpande S, Watson L & Canfield R, 2013, *Pareto front approximation using a hybrid approach*, Procedia Computer Science, **18**, pp. 521–530.

[45] Diaz-Gomez PA & Hougen DF, 2007, *Initial population for genetic algorithms: A metric approach*, Proceedings of the 2007 International Conference on Genetic and Evolutionary Methods, pp. 43–49.

[46] Diffen, 2016, *Goal vs. Objective*, [Online], [Cited February 15^{th}, 2015], Available from `http://www.diffen.com/difference/Goal_vs_Objective`.

[47] Dorigo M & Gambardella LM, 1997, *Ant colony system: A cooperative learning approach to the traveling salesman problem*, IEEE Transactions on Evolutionary Computation, **1(1)**, pp. 53–66.

[48] Dréo J, 2006, *Metaheuristics for hard optimization: Methods and case studies*, Springer, Paris.

[49] Du Toit J, 2014, *Decision support for threat detection in maritime surveillance*, PhD Dissertation, University of Stellenbosch, Stellenbosch.

[50] Eberhart RC & Kennedy J, 1995, *A new optimizer using particle swarm theory*, Proceedings of the 6^{th} International Symposium on Micro Machine and Human Science, **1**, pp. 39–43.

[51] Eglese R, 1990, *Simulated annealing: A tool for operational research*, European Journal of Operational Research, **46(3)**, pp. 271–281.

[52] Encyclopaedia Britannica, 2015, *Law of large numbers*, [Online], [Cited March 22nd, 2016], Available from `http://global.britannica.com/science/law-of-large-numbers`.

[53] EPFL, 2015, *Conditional Probability*, [Online], [Cited March 14th, 2016], Available from `http://statwww.epfl.ch/davison/teaching/ProbStatSC/20032004/week4and5.pdf`.

[54] Farahbod R, Glasser U & Khalili A, 2009, *A multi-layer network architecture for dynamic resource configuration and management of multiple mobile resources in maritime surveillance*, International Society for Optics and Photonics, **7345**, pp. 734508–734508.

[55] Farina A, Giompapa S, Gini F, Graziano A & Stefano RD, 2007, *Computer simulation of an integrated multi-sensor system for maritime border control*, Proceedings of the 2007 IEEE Radar Conference, pp. 308–313.

[56] Farina M, Deb K & Amato P, 2004, *Dynamic multiobjective optimization problems: Test cases, approximations, and applications*, IEEE Transactions on Evolutionary Computation, **8(5)**, pp. 425–442.

[57] Fisheries and Oceans Canada, 2013, *Canada to extend life of coast guard fleet*, [Online], [Cited February 22nd, 2013], Available from `http://www.worldmaritimenews.com/archives/77223/canada-to-extend-life-of-coast-guard-fleet/`.

[58] Fonseca CM, Knowles JD, Thiele L & Zitzler E, 2005, *A tutorial on the performance assessment of stochastic multiobjective optimizers*, Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization, pp. 240–249.

[59] Fry F, 2009, *gCaptain*, [Online], [Cited November 16th, 2013], Available from `http://gcaptain.com/?p=11136`.

[60] Gad A & Farooq M, 2002, *Data fusion architecture for maritime surveillance*, Proceedings of the 5th IEEE International Conference on Information Fusion, **1**, pp. 448–455.

[61] Gass SI & Harris CM, 2001, *Encyclopedia of operations research and management science: Centennial edition*, Springer Science & Business Media, New York (NY).

[62] Giompapa S, Gini F, Farina A, Graziano A & Di Stefano R, 2009, *Maritime border control multisensor system*, IEEE Aerospace and Electronic Systems Magazine, **24(8)**, pp. 9–15.

[63] Global Economics, 2012, *Why China can't afford a confrontation with Japan*, [Online], [Cited September 15th, 2013], Available from `http://www.businessweek.com/articles/2012-08-20/why-china-cant-afford-a-confrontation-with-japan`.

[64] Golden BL & Stewart W, 1978, *Vehicle routing with probabilistic demands*, Proceedings of the Journal of Computer Science and Statistics: Tenth Annual Symposium on the Interface, Gaithersburg (MD), pp. 252–259.

[65] Herrera-Viedma E, Herrera F & Chiclana F, 2002, *A consensus model for multiperson decision making with different preference structures*, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, **32(3)**, pp. 394–402.

[66] Ho W, Ho GT, Ji P & Lau HC, 2008, *A hybrid genetic algorithm for the multi-depot vehicle routing problem*, Journal of Engineering Applications of Artificial Intelligence, **21(4)**, pp. 548–557.

[67] Ho W & Ji P, 2003, *Component scheduling for chip shooter machines: A hybrid genetic algorithm approach*, Computers and Operations Research, **30(14)**, pp. 2175–2189.

[68] Holland JH, 1975, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, University of Michigan Press, Ann Arbor (MI).

[69] Horn J, Nafploitis N & Goldberg D, 1994, *A niched Pareto genetic algorithm for multi-objective optimization*, Proceedings of the 1999 IEEE International Conference on Evolutionary Computation, pp. 82–87.

[70] Israel Homeland Security Ground, 2014, *Northrop Grumman develops innovative marine unmanned system*, [Online], [Cited May 15th, 2014], Available from `http://i-hls.com/2014/10/northrop-grumman-unveils-innovative-marine-unmanned-system/`.

[71] Jacobson L, 2013, *Simulated annealing for beginners*, [Online], [Cited June 12th, 2013], Available from `http://www.theprojectspot.com/tutorial-post/simulated-annealing-algorithm-for-beginners/6`.

[72] Jahn J & Krabs W, 2012, *Recent advances and historical development of vector optimization*, Springer Science & Business Media, Berlin.

[73] Jain AK, Murty MN & Flynn PJ, 1999, *Data clustering: A review*, ACM Computing Surveys, **31(3)**, pp. 264–323.

[74] Jin Y & Branke J, 2005, *Evolutionary optimization in uncertain environments: A survey*, IEEE Transactions on Evolutionary Computation, **9(3)**, pp. 303–317.

[75] Kapadia AS, Chan W & Moyé LA, 2005, *Mathematical statistics with applications*, CRC Press, Boca Raton (FL).

[76] Kennedy J & Eberhart R, 1995, *Particle swarm optimization*, Swarm Intelligence, **4**, pp. 1942–1948.

[77] Kettani H & Ostrouchov G, 2005, *On the distribution of the distance between two multivariate normally distributed points*, Department of Computer Science and Information Systems Engineering, Fort Hays State University, Fort Hays (KS).

[78] Kirkpatrick S, Gelatt C & Vecchi M, 1984, *Optimization by simulated annealing: Quantitative studies*, Journal of Statistical Physics, **34(5)**, pp. 975–986.

[79] Knowles JD & Corne DW, 2000, *Approximating the nondominated front using the Pareto archived evolution strategy*, The Journal of Evolutionary Computation, **8(2)**, pp. 149–172.

[80] Kokubugata H & Kawashima H, 2008, *Application of simulated annealing to routing problems in city logistics*, Journal of Simulated Annealing, **40**, pp. 420–430.

[81] Konak A, Coit D & Smith A, 2006, *Multi-objective optimization using genetic algorithms: A tutorial*, Reliability Engineering and System Safety, **91(9)**, pp. 992–1007.

[82] Lane RO, Nevell DA, Hayward SD & Beaney TW, 2010, *Maritime anomaly detection and threat assessment*, Proceedings of the 13th Conference on Information Fusion, pp. 1–8.

[83] Laporte G, Louveaux F & Mercure H, 1992, *The vehicle routing problem with stochastic travel times*, Journal of Transportation Science, **26(3)**, pp. 161–170.

[84] LaTeX Stack Exchange, 2014, *Geometry of over-constrainedness*, [Online], [Cited April 6th, 2016], Available from `http://tex.stackexchange.com/questions/158668/nice-scientific-pictures-show-off/212936`.

[85] Laxhammar R, 2008, *Anomaly detection for sea surveillance*, Proceedings of the 11th International Conference on Information Fusion, pp. 1–8.

[86] Lee D & Schachter B, 1980, *Two algorithms for constructing a Delaunay triangulation*, International Journal of Computer & Information Sciences, **9(3)**, pp. 219–242.

[87] Lin S, Yu VF & Lu C, 2011, *A simulated annealing heuristic for the truck and trailer routing problem with time windows*, Expert Systems with Applications, **38(12)**, pp. 15244–15252.

[88] Liu S, Huang W & Ma H, 2009, *An effective genetic algorithm for the fleet size and mix vehicle routing problems*, Journal of Transportation Research Part E: Logistics and Transportation Review, **45(3)**, pp. 434–445.

[89] Lötter DP and van Vuuren JH, 2014, *Weapon assignment decision support in a surface-based air defence environment*, Military Operations Research, Submitted.

[90] Lötter DP, 2012, *Modelling weapon assignment as a multiobjective decision problem*, MSc Thesis, University of Stellenbosch, Stellenbosch.

[91] Malik A, Maciejewski R, Jang Y, Oliveros S, Yang Y, Maule B, White M & Ebert DS, 2011, *A visual analytics process for maritime resource allocation and risk assessment*, Proceedings of the 2011 IEEE Conference on Visual Analytics Science and Technology, pp. 221–230.

[92] Marine Defenders, 2012, *Marine defenders project*, [Online], [Cited April 24th, 2013], Available from `http://www.marinedefenders.com/oilpollutionfacts/illegal.php`.

[93] Martin GE, 2001, *Counting: The art of enumerative combinatorics*, Springer Science & Business Media, New York (NY).

[94] Maryland C, 2012, *US Coast Guard: Seaboard marine guardians*, [Online], [Cited October 16th, 2013], Available from `http://us-coastguard.blogspot.com/`.

[95] Mathematics Stack Exchange, 2011, *Multivariate normal difference distribution*, [Online], [Cited February 12th, 2016], Available from `http://math.stackexchange.com/questions/60911/multivariate-normal-difference-distribution`.

[96] Mathematics Stack Exchange, 2011, *Percentage of overlapping regions of two normal distributions*, [Online], [Cited March 12th, 2016], Available from `http://stats.stackexchange.com/questions/12209/percentage-of-overlapping-regions-of-two-normal-distributions`.

[97] Mathematics Stack Exchange, 2014, *Product of two multivariate Gaussian distributions*, [Online], [Cited February 13th, 2016], Available from `http://math.stackexchange.com/questions/157172/product-of-two-multivariate-gaussians-distributions`.

[98] McLachlan GJ, 1999, *Mahalanobis distance*, Resonance, **4(6)**, pp. 20–26.

[99] MEDESS, 2013, *Mediterranean decision support system for marine safety*, [Online], [Cited June 5th, 2013], Available from `http://www.medess4ms.eu/emergencies`.

[100] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH & Teller E, 1953, *Equation of state calculations by fast computing machines*, The Journal of Chemical Physics, **21(6)**, pp. 1087–1092.

[101] Montemanni R, Gambardella LM, Rizzoli AE & Donati AV, 2005, *Ant colony system for a dynamic vehicle routing problem*, Journal of Combinatorial Optimization, **10(4)**, pp. 327–343.

[102] Moore K, 2005, *Predictive analysis of naval deployment activities*, [Online], [Cited February 13th, 2013], Available from `http:www.fbo.gov/utils/view?id=4b9eec01c0278620e50704ed51e24785`.

[103] Moreira JE & Naik VK, 1997, *Dynamic resource management on distributed systems using reconfigurable applications*, IBM Journal of Research and Development, **41(3)**, pp. 303–330.

[104] Moskowitz H & Tang J, 2000, *Multiattribute utility models*, Journal of Decision Sciences, **31(2)**.

[105] Murata T & Ishibuchi H, 1995, *MOGA: Multi-objective genetic algorithms*, Proceedings of the 1995 IEEE International Conference on Evolutionary Computation, pp. 289–298.

[106] Nazif H & Lee LS, 2010, *Optimized crossover genetic algorithm for vehicle routing problem with time windows*, American Journal of Applied Sciences, **7(1)**, pp. 95–101.

[107] Neural Networks and Pattern Recognition Tutorial, 2015, *Bayesian decision theory*, [Online], [Cited January 12th, 2016], Available from `https://www.byclb.com/TR/Tutorials/neural_networks/ch4_1.htm`.

[108] Norozi A, Ariffin M, Ismail N & Mustapha F, 2011, *An optimization technique using hybrid GA-SA algorithm for multi-objective scheduling problem*, Journal of Scientific Research and Essays, **6(8)**, pp. 1720–1731.

[109] Ochi LS, Vianna DS, Drummond LM & Victor AO, 1998, *A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet*, Series Lecture Notes in Computer Science, **1388**, pp. 216–224.

[110] Parsons L, Haque E & Liu H, 2004, *Subspace clustering for high dimensional data: A review*, ACM SIGKDD Explorations Newsletter, **6(1)**, pp. 90–105.

[111] Pascoa S, 2012, *The sunken billions: The economic justification of fisheries reform*, Marine Resource Economics, **27(2)**, pp. 193–194.

[112] Petersen KB & Pedersen MS, 2008, *The matrix cookbook*, Technical University of Denmark, Copenhagen.

[113] Pillac V, Gendreau M, Guéret C & Medaglia AL, 2013, *A review of dynamic vehicle routing problems*, European Journal of Operational Research, **225(1)**, pp. 1–11.

[114] Poli R, Kennedy J & Blackwell T, 2007, *Particle swarm optimization: An overview*, Proceedings of the 1995 IEEE International Conference on Neural Networks, **1(1)**, pp. 33–57.

[115] Ponnambalam S & Reddy M, 2003, *A GA-SA multiobjective hybrid search algorithm for integrating lot sizing and sequencing in flow-line scheduling*, The International Journal of Advanced Manufacturing Technology, **21(2)**, pp. 126–137.

[116] Ponsford AM, Sevgi L & Chan HC, 2001, *An integrated maritime surveillance system based on high-frequency surface-wave radars, Part 2: Operational status and system performance*, IEEE Antennas and Propagation Magazine, **43(5)**, pp. 52–63.

[117] Powell A, 1966, *A complete system of consumer demand equations for the Australian economy fitted by a model of additive preferences*, Econometrica: Journal of the Econometric Society, **34(3)**, pp. 661–675.

[118] Psaraftis H, 1995, *Dynamic vehicle routing: Status and prospects*, Annals of Operations Research, **61**, pp. 143–164.

[119] Puljić K & Manger R, 2013, *Comparison of eight evolutionary crossover operators for the vehicle routing problem*, Journal of Mathematical Communications, **18(2)**, pp. 359–375.

[120] Raad DN, 2011, *Multiobjective optimisation of water distribution systems design using metaheuristics*, PhD Dissertation, University of Stellenbosch, Stellenbosch.

[121] Roberts S, Osborne M, Ebden M, Reece S, Gibson N & Aigrain S, 2013, *Gaussian processes for time-series modelling*, Philosophical Transactions of the Royal Society of London: Mathematical, Physical and Engineering Sciences, **371(1984)**, pp. 550–560.

[122] Rosen K & Michaels J, 2000, *Handbook of discrete and combinatorial mathematics*, CRC Press LLC, Boca Raton (FL).

[123] Salhi S, Imran A & Wassan NA, 2009, *The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation*, [Online], [Cited March 6th, 2014], Available from `http://www.kent.ac.uk/kbs/documents/res/working-papers/2013/mdvfm%20paper(May%202013)%20Web.pdf`.

[124] Sandoz JF, 2012, *Maritime security sector reform*, US Institute of Peace, Washington (DC).

[125] Scheepens R, Willems N, van de Wetering H and van Wijk JJ, 2011, *Interactive visualization of multivariate trajectory data with density maps*, Proceedings of the 2011 IEEE Pacific Visualization Symposium, pp. 147–154.

[126] Schlünz EB, 2014, University of Stellenbosch, [Personal Communication], Contactable at `bernard.schlunz@necsa.co.za`.

[127] Serafini P, 1994, *Simulated annealing for multi objective optimization problems*, Springer, New York (NY).

[128] Shahsavari-Pour N & Ghasemishabankareh B, 2013, *A novel hybrid meta-heuristic algorithm for solving multi objective flexible job shop scheduling*, Journal of Manufacturing Systems, **32(4)**, pp. 771–780.

[129] Shieh E, An B, Yang R, Tambe M, Baldwin C, DiRenzo J, Maule B and Meyer G, 2012, *PROTECT: A deployed game theoretic system to protect the ports of the United States*, Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, pp. 13–20.

[130] Smith KI, Everson RM & Fieldsend JE, 2004, *Dominance measures for multiobjective simulated annealing*, IEEE Transactions on Evolutionary Computation, **1**, pp. 23–30.

[131] Smith KI, Everson RM, Fieldsend JE, Murphy C & Misra R, 2008, *Dominance-based multiobjective simulated annealing*, IEEE Transactions on Evolutionary Computation, **12(3)**, pp. 323–342.

[132] Smith RC & Cheeseman P, 1986, *On the representation and estimation of spatial uncertainty*, The International Journal of Robotics Research, **5(4)**, pp. 56–68.

[133] Spruyt V, 2014, *How to draw a covariance error ellipse?*, [Online], [Cited December 15th, 2015], Available from `http://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix/`.

[134] Srinivas N & Deb K, 1994, *Muiltiobjective optimization using nondominated sorting in genetic algorithms*, IEEE Transactions on Evolutionary computation, **2(3)**, pp. 221–248.

[135] Stankovianska I, 2006, *An effective approach to routing problems*, Scientific Papers of the University of Pardubice, Faculty of Economics and Administration, Zilina.

[136] Stewart WR & Golden BL, 1983, *Stochastic vehicle routing: A comprehensive approach*, European Journal of Operational Research, **14(4)**, pp. 371–385.

[137] Stroupe AW, Martin MC & Balch T, 2000, *Merging probabilistic observations for mobile distributed sensing*, (Unpublished) Technical Report, The Robotics Institute, Carnegie Mellon University, Pittsburgh (PA).

[138] Suman B & Kumar P, 2006, *A survey of simulated annealing as a tool for single and multiobjective optimization*, Journal of the Operational Research Society, **57(10)**, pp. 1143–1160.

[139] Suppapitnarm A, Seffen K, Parks G & Clarkson P, 2000, *A simulated annealing algorithm for multiobjective optimization*, Journal of Engineering Optimization, **33(1)**, pp. 59–85.

[140] Surekha P & Sumathi S, 2011, *Solution to multi-depot vehicle routing problem using genetic algorithms*, Journal of World Applied Programming, **1(3)**, pp. 118–131.

[141] Szymon J & Dominik Z, 2013, *Solving multi-criteria vehicle routing problem by parallel tabu search on GPU*, Procedia Computer Science, **18**, pp. 2529–2532.

[142] Tabassum M & Mathew K, 2014, *A genetic algorithm analysis towards optimization solutions*, International Journal of Digital Information and Wireless Communications, **4(1)**, pp. 124–142.

[143] Tahernezhadiani K, Hamzeh A & Hashemi S, 2012, *Towards enhancing solution space diversity in multi objective optimisation: A hypervolume-based approach*, International Journal of Artificial Intelligence & Applications, **3(1)**, pp. 65–81.

[144] Tanaka M, 2012, *Product of two multivariate Gaussians distributions*, [Online], [Cited February 13th, 2016], Available from `http://math.stackexchange.com/questions/157172/product-of-two-multivariate-gaussians-distributions`.

[145] Tao Y, Papadias D & Shen Q, 2002, *Continuous nearest neighbor search*, Proceedings of the 28*th* International Conference on Very Large Data Bases, pp. 287–298.

[146] The Guardian, 2011, *Al-Qaida hoped to blow up oil tankers*, [Online], [Cited February 28th, 2016], Available from `http://www.theguardian.com/world/2011/may/20/al-qaida-oil-tankers-bin-laden`.

[147] The MathWorks, 2010, *MATLAB version 7.10.0 (R2010a)*, Natick (MA).

[148] The Week, 2014, *Al Qaeda's next target: American oil tankers*, [Online], [Cited February 28th, 2016], Available from `http://theweek.com/articles/442842/al-qaedas-next-target-american-oil-tankers`.

[149] Thietart R & Forgues B, 1995, *Chaos theory and organization*, Organization Science, **6(1)**, pp. 19–31.

[150] Toth P & Vigo D, 2005, *The vehicle routing problem*, SIAM Monographs on Discrete Mathematics and Applications, Camden (PA).

[151] Treves T, 2008, *United Nations Convention on the Law of the Sea*, [Online], [Cited April 29th, 2013], Available from `https://www.un.org/depts/los/convention_agreements/texts/unclos/UNCLOS-TOC.htm`.

[152] Triki E, Collette Y & Siarry P, 2005, *A theoretical study on the behavior of simulated annealing leading to a new cooling schedule*, European Journal of Operational Research, **166(1)**, pp. 77–92.

[153] UK Essays, 2013, *Enumeration deterministic heuristic and stochastic optimisation computer science essay*, [Online], [Cited October 2nd, 2014], Available from `http://www.ukessays.com/essays/computer-science/enumeration-deterministic-heuristic-and-stochastic-optimisation-computer-science-essay.php`.

[154] Ulrich T, Bader J & Zitzler E, 2010, *Integrating decision space diversity into hypervolume-based multiobjective search*, Proceedings of the 12^{th} Annual Conference on Genetic and Evolutionary Computation, pp. 455–462.

[155] Universe Today: Space and Astronomy News, 2008, *Arecibo joins forces with Global Antennae to simulate 6,800 mile telescope*, [Online], [Cited November 10th, 2015], Available from `http://www.universetoday.com/14969/arecibo-joins-forces-with-global-antennae-to-simulate-6800-mile-telescope/`.

[156] van Breedam A, 1995, *Improvement heuristics for the vehicle routing problem based on simulated annealing*, European Journal of Operational Research, **86(3)**, pp. 480–490.

[157] Vidal T, Crainic TG, Gendreau M, Lahrichi N & Rei W, 2012, *A hybrid genetic algorithm for multidepot and periodic vehicle routing problems*, Operations Research, **60(3)**, pp. 611–624.

[158] Wehn H, Yates R, Valin P, Guitouni A, Bossé É, Dlugan A & Zwick H, 2007, *A distributed information fusion testbed for coastal surveillance*, Proceedings of the 10th International Conference on Information Fusion, pp. 1–7.

[159] While L, Bradstreet L, Barone L & Hingston P, 2005, *Heuristics for optimizing the calculation of hypervolume for multi-objective optimization problems*, Proceedings of the 2005 IEEE Congress on Evolutionary Computation, pp. 2225–2232.

[160] While L, Hingston P, Barone L & Huband S, 2006, *A faster algorithm for calculating hypervolume*, IEEE Transactions on Evolutionary Computation, **10(1)**, pp. 29–38.

[161] Wiener N, 1949, *Extrapolation, interpolation, and smoothing of stationary time series*, MIT press Cambridge (MA).

[162] Woldemariam KM, 2008, *Multimodal and constrained optimization in artificial immune system*, MSc Thesis, Oklahoma State University, Stillwater (OK).

[163] Wolfram Mathworld, 2015, *Bivariate normal distribution*, [Online], [Cited October 27th, 2015], Available from `http://mathworld.wolfram.com/BivariateNormalDistribution.html`.

[164] Wu T, Low C & Bai J, 2002, *Heuristic solutions to multi-depot location-routing problems*, Computers and Operations Research, **29(10)**, pp. 1393–1415.

[165] Xiaohui H, 2006, *Particle swarm optimization*, [Online], [Cited September 27th, 2014], Available from `http://www.swarmintelligence.org/index.php`.

[166] Yang S, 2005, *Population-based incremental learning with memory scheme for changing environments*, Proceedings of the 7^{th} Annual Conference on Genetic and Evolutionary Computation, pp. 711–718.

[167] Zitzler E, Deb K & Thiele L, 2000, *Comparison of multiobjective evolutionary algorithms: Empirical results*, IEEE Transactions on Evolutionary Computation, **8(2)**, pp. 173–195.

[168] Zitzler E, Brockhoff D & Thiele L, 2000, *The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration*, The Journal of Evolutionary Computation, **8(2)**, pp. 173–195.

[169] Zitzler E & Thiele L, 1999, *Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach*, IEEE Transactions on Evolutionary Computation, **3(4)**, pp. 257–271.

APPENDIX A

Preliminary experiments

This appendix contains the results obtained from the preliminary experiments conducted using the method of simulated annealing and the NSGA-II to solve the lower-complexity hypothetical scenario of § 8.3. The reason behind conducting these experiments was to acquire an understanding of suitable parameter values for each search method in order to be able to set up the main experiments in a more effective manner.

Exp	Perturbation parameters						Cooling			1-minute run			2-minute runs		
#	p_{wr}	p_{brs}	p_{brd}	p_{is}	α	β	T_0	π	C	$\bar{I}$	$\lvert\bar{\mathcal{A}}\rvert$	$\bar{H}^i$	$\bar{I}$	$\lvert\bar{\mathcal{A}}\rvert$	$\bar{H}^i$
1	0.5	0.1	0.1	0.3	1.5	1	5	0.82	30	24 654	329	16 284	52 258	453	40 411
2	0.1	0.2	0.2	0.5	1.5	1	5	0.82	30	23 910	350	23 234	49 571	478	44 569
3	0.4	0.1	0.1	0.4	2.5	1.5	5	0.82	30	23 670	320	21 209	48 383	449	34 683
4	0.1	0.2	0.2	0.5	2	1	10	0.82	30	23 480	325	16 240	49 040	406	25 803
5	0.3	0.15	0.15	0.4	1.5	1	2	0.82	30	24 148	372	21 644	48 690	519	56 132
6	0.15	0.2	0.2	0.45	2	1	3.5	0.85	30	23 848	304	16 285	49 822	456	37 481
7	0.05	0.15	0.15	0.65	2	1.5	5	0.85	20	23 045	282	17 971	46 093	358	26 584
8	0.1	0.2	0.2	0.5	2	1	7	0.85	40	23 650	362	23 497	46 969	425	31 041
9	0.1	0.15	0.15	0.6	2.5	1.5	5	0.85	25	23 165	287	15 739	48 295	370	23 843
10	0.05	0.15	0.15	0.65	1	1.5	4	0.85	40	22 772	390	31 185	45 986	447	47 474
11	0.1	0.15	0.15	0.6	1	2	5	0.85	50	22 141	404	39 795	46 109	556	79 856
12	0.05	0.15	0.15	0.65	1.5	2	7	0.85	100	22 083	519	60 155	43 192	688	128 141
13	0.2	0.1	0.1	0.6	2	2	10	0.9	150	22 056	452	48 065	44 282	677	108 218
14	0.15	0.15	0.15	0.55	2.5	1.5	30	0.9	250	22 799	573	66 479	45 193	795	144 438
15	0.15	0.15	0.15	0.55	2.5	1.5	30	0.85	50	23 180	324	20 624	46 430	440	46 148
16	0.15	0.15	0.15	0.55	2.5	1.5	15	0.9	90	23 079	367	28 647	46 116	542	56 134
17	0.25	0.15	0.15	0.45	3	1	30	0.85	200	24 037	558	63 315	48 315	854	150 730
18	0.2	0.1	0.1	0.6	2.5	1.5	30	0.85	150	22 136	522	61 686	44 621	751	115 602
19	0.2	0.15	0.15	0.5	2.5	1.5	30	0.9	300	23 201	547	63 507	46 857	780	143 104
20	0.25	0.1	0.1	0.55	3	1.5	30	0.9	250	22 842	533	67 216	46 056	764	140 518
21	0.2	0.15	0.15	0.5	5	3	30	0.9	250	22 082	488	69 377	44 025	666	130 156

TABLE A.1: *Results of the preliminary experiments conducted using the method of simulated annealing to solve the lower-complexity hypothetical scenario of § 8.3.*

Exp	Perturbation parameters						Cooling			1-minute run			2-minute runs		
#	p_{wr}	p_{brs}	p_{brd}	p_{is}	α	β	T_0	π	C	$\bar{I}$	$\lvert\bar{\mathcal{A}}\rvert$	$\bar{H}^i$	$\bar{I}$	$\lvert\bar{\mathcal{A}}\rvert$	$\bar{H}^i$
22	0.2	0.15	0.15	0.5	1.5	1	30	0.85	200	23 781	637	75 442	48 115	830	130 497
23	0.5	0.1	0.1	0.3	1.5	2.5	30	0.85	150	24 454	381	38 975	49 358	582	85 849
24	0.1	0.2	0.2	0.5	2	2.5	40	0.9	100	23 687	373	29 789	47 033	531	75 610
25	0.1	0.2	0.2	0.5	1	2.5	30	0.9	150	23 052	453	49 757	46 164	622	92 927
26	0.1	0.2	0.2	0.5	2	2	30	0.85	200	23 687	550	80 666	47 680	799	146 152
27	0.3	0.1	0.1	0.5	2	2	30	0.85	200	24 057	540	62 047	48 115	776	139 868
28	0.2	0.15	0.15	0.5	2	2.5	30	0.85	200	23 935	533	63 121	47 972	819	165 128
29	0.2	0.15	0.15	0.5	3.5	3	30	0.85	200	23 259	494	57 089	46 380	694	114 219
30	0.2	0.15	0.15	0.5	4.5	4	30	0.85	200	22 842	444	64 028	45 723	687	158 910
31	0.2	0.15	0.15	0.5	5.5	5	30	0.85	200	22 285	430	74 497	45 064	737	170 324
32	0.2	0.15	0.15	0.5	6.5	6	30	0.85	200	22 514	452	75 284	40 598	611	132 679
33	0.2	0.15	0.15	0.5	7.5	7	30	0.85	200	22 473	436	73 352	44 950	661	156 220
34	0.2	0.15	0.15	0.5	9	9	30	0.85	200	23 431	456	24 200	45 320	638	167 712
35	0.2	0.15	0.15	0.5	11	11	30	0.85	200	22 551	442	74 163	45 041	680	148 474
36	0.2	0.15	0.15	0.5	15	13	30	0.85	200	23 184	451	96 508	46 388	644	207 742
37	0.2	0.15	0.15	0.5	18	16	30	0.85	200	23 197	419	93 519	46 199	619	171 418
38	0.2	0.15	0.15	0.5	24	22	30	0.85	200	23 600	404	84 463	46 579	583	141 408
39	0.2	0.15	0.15	0.5	30	28	30	0.85	200	23 573	439	95 030	47 378	555	153 720
40	0.2	0.15	0.15	0.5	100	100	30	0.85	200	23 651	388	84 492	47 594	525	153 890
41	0.75	0	0	0.25	5.5	5	30	0.85	200	24 490	274	22 182	49 474	355	38 557
42	0	0	0	1	5.5	5	30	0.85	200	19 678	359	40 181	38 381	475	67 165

TABLE A.2: *Results of the preliminary experiments conducted using the method of simulated annealing to solve the lower-complexity hypothetical scenario of § 8.3 (continued).*

Exp	Perturbation parameters						Cooling			1-minute run			2-minute runs		
#	p_{wr}	p_{brs}	p_{brd}	p_{is}	α	β	T_0	π	C	$\bar{I}$	$\lvert\bar{\mathcal{A}}\rvert$	$\bar{H}^i$	$\bar{I}$	$\lvert\bar{\mathcal{A}}\rvert$	$\bar{H}^i$
43	0	0.35	0.35	0.3	5.5	5	30	0.85	200	23 570	559	118 329	48 134	663	159 395
44	0	0.45	0.45	0.1	5.5	5	30	0.85	200	26 187	474	74 848	52 366	684	164 042
45	0	0.8	0	0.2	5.5	5	30	0.85	200	23 968	285	30 291	48 842	380	46 935
46	0	0	0.8	0.2	5.5	5	30	0.85	200	25 154	530	112 360	51 201	704	149 012
47	0	0.25	0.55	0.2	5.5	5	30	0.85	200	25 264	531	104 812	52 718	586	129 027
48	0	0.55	0.25	0.2	5.5	5	30	0.85	200	24 727	478	94 684	50 155	638	128 479
49	0	0.45	0.35	0.2	5.5	5	30	0.85	200	24 321	515	100 111	50 719	655	167 182
50	0	0.35	0.45	0.2	5.5	5	30	0.85	200	24 782	455	68 280	51 104	691	174 786
51	0.4	0.3	0.3	0	5.5	5	30	0.85	200	22 429	300	49 652	45 386	464	106 481
52	0.2	0.1	0.15	0.55	5.5	5	30	0.99	300	22 967	169	6 034	46 457	216	12 252
53	0.2	0.1	0.15	0.55	5.5	5	30	0.92	300	22 097	437	76 847	44 193	657	171 258
54	0.2	0.1	0.15	0.55	5.5	5	30	0.92	500	21 920	501	73 069	43 581	625	122 732
55	0.2	0.1	0.15	0.55	5.5	5	30	0.92	100	22 966	261	22 838	45 911	350	34 938
56	0.2	0.1	0.15	0.55	5.5	5	30	0.92	700	21 518	495	77 906	44 397	762	184 968
57	0.2	0.1	0.15	0.55	5.5	5	10	0.92	150	22 876	390	47 667	45 318	534	109 747
58	0.2	0.1	0.15	0.55	5.5	5	30	0.85	300	21 718	516	89 810	44 513	716	166 746
59	0.2	0.1	0.15	0.55	5.5	5	30	0.92	900	21 572	496	91 235	42 792	651	148 067
60	0.2	0.1	0.15	0.55	5.5	5	30	0.92	1200	21 431	534	83 092	42 413	696	179 338
61	0.2	0.1	0.15	0.55	5.5	5	30	0.85	250	22 222	504	80 948	44 480	612	150 953
62	0.2	0.1	0.15	0.55	5.5	5	30	0.85	700	21 609	503	93 862	43 287	766	214 954
63	0.2	0.1	0.15	0.55	5.5	5	30	0.85	800	21 617	531	79 053	43 170	728	195 722

TABLE A.3: *Results of the preliminary experiments conducted using the method of simulated annealing to solve the lower-complexity hypothetical scenario of § 8.3 (continued).*

Exp	Mutation rate	Population parameters				1-minute run			2-minute runs		
#	p_m	σ	μ	λ	P_{size}	$\bar{G}$	$\lvert\bar{\mathcal{F}}_1\rvert$	$\bar{H}^i$	$\bar{G}$	$\lvert\bar{\mathcal{F}}_1\rvert$	$\bar{H}^i$
1	0.1	1.5	1	20	548	84	297	15 014	164	302	17 893
2	0.1	2	1.5	30	1 118	39	612	79 140	77	613	73 183
3	0.1	4	2	10	3 254	12	489	34 853	25	1 515	191 928
4	0.1	1	2.5	5	2 234	17	624	67 813	37	1 220	183 840
5	0.1	10	1	10	1 262	35	687	85 530	68	691	99 593
6	0.1	20	1	10	2 342	18	758	70 019	37	1 282	207 558
7	0.1	0	1	30	3 242	12	606	43 431	25	1 809	257 746
8	0.1	0	1	100	1 802	25	997	98 887	50	982	180 978
9	0.1	0	1	75	1 352	33	736	93 791	65	734	111 280
10	0.1	0	1	125	2 252	20	1 304	125 088	40	1 214	221 360
11	0.1	35	1	0	3 782	10	512	33 441	21	1 407	175 262
12	0.1	40	1	0	4 322	9	482	32 480	19	1 232	158 054
13	0.1	25	1	0	2 702	16	696	60 314	32	1 504	223 142
14	0.1	1	2	0	770	56	427	46 001	112	424	51 699
15	0.1	1	2.2	0	1 166	37	631	73 082	74	629	109 725
16	0.1	1	2.4	0	1 748	23	850	72 891	49	940	179 574
17	0.1	1	2.6	0	2 636	14	445	38 410	31	1 352	205 548
18	0.05	2	1.5	30	1 118	41	608	77 563	80	620	83 902
19	0	25	1	5	1396	15	761	65 862	31	1 601	228 048
20	0.05	25	1	5	1396	15	723	48 449	30	1 586	264 056
21	0.1	25	1	5	1396	15	598	52 357	31	1 590	231 838
22	0.15	25	1	5	1396	14	711	51 500	30	1 600	205 046
23	0.2	25	1	5	1396	15	686	59 024	30	1 553	227 164
24	0.3	25	1	5	1396	15	711	71 836	31	1 585	311 758

TABLE A.4: *Results of the preliminary experiments conducted using the NSGA-II to solve the lower-complexity hypothetical scenario of § 8.3.*